Design • Build • Program • Learn • Compete

SERVO MAGAZINE

FOR THE ROBOT INNOVATOR

www.servomagazine.com

December 2011

PR LITE

Four High School Roboticists Roll Their Own PR2

✦ Wonder Wheels

Put a new spin on your robot base with omni or mecanum style wheels.

✦ Pimp My Hexapod

A Frankenstein mixture of parts is transformed into a state-of-the-art six-legged walker.


LISTEN UP!
YOUR BOT IS TALKING!
Knowing what your robot is doing can make a world of difference when you are competing. Our HTS-SS Blue Sensor station and series of sensors puts valuable GPS, speed, altitude, RPM, temperature, voltage and current information at your fingertips. Couple our HTS-Voice with your Hitec radio and your bot will actually tell you the data you want to hear. Plug the HTS-Navi into your PC and analyze your data at the end of the day!
HITEC 2.4GHz
ADAPTIVE FREQUENCY HOPPING SPREAD SPECTRUM
GET EQUIPPED!
GET EMPOWERED!
GET HITEC!
HTS-VOICE
HTS-SS BLUE
HTS-NAVI
[Current Sensor]
Current : 95.2A / C200
Wattage : 1332.8W
Warning (Hi) : 100A
3/3
MAX 98.0A
MAX 1421W
RST
AURORA 9
AURORA 9
Installed with the
HTS-VOICE
HITEC
12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecrcd.com

SERVO MAGAZINE

12.2011
VOL. 9 NO. 12

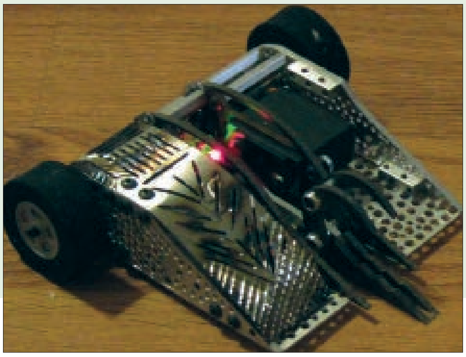
PAGE 08

PAGE 10

The Combat Zone...

Features

Events



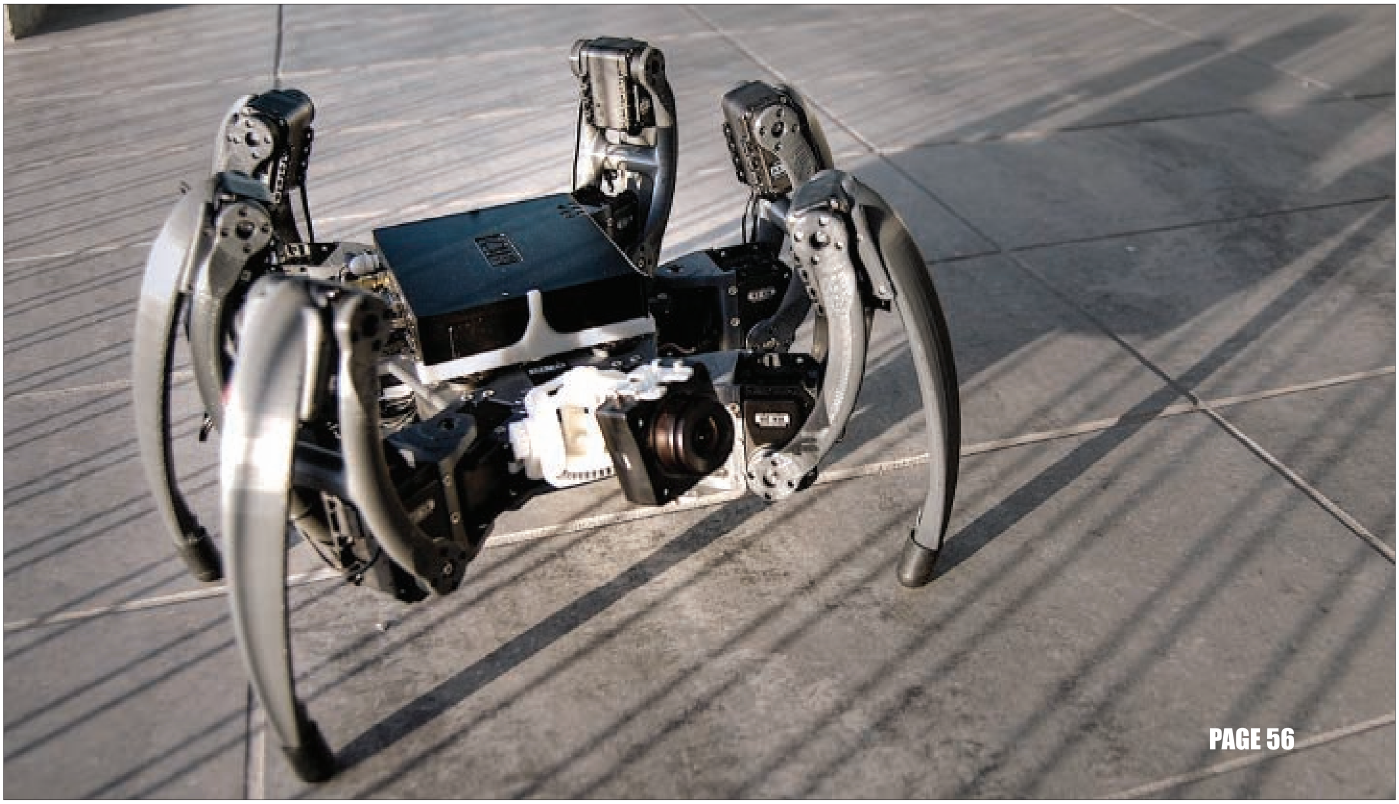
Columns

Departments

PAGE 24

SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

In This Issue ...



Mind / Iron

by Bryan Bergeron, Editor

Moongel, Drum Dials, and Aurora Strings

Experimenters who have spent significant time designing and building robots know that often the greatest challenge is locating affordable parts. For example, oftentimes I know exactly the sort of sensor I need, but can't seem to find it in the online robotic shops. When pressed for time, I sometimes cannibalize a perfectly good working robot for a $50 sensor. As a result, my collection of bots resembles a hospital ward filled with hobbled patients.

Time and budget allowing, I have occasionally succumbed to the lure of the specialty electronic supply companies. At the top of my list is Omega Measurement (**www.omega.com**) which sells high quality sensors for anything imaginable. Just be prepared to pay for it. My collection of Omega catalogs and 'how to measure' books is so useful that it commands an entire shelf in my bookcase. In addition to providing an industry-recognized name for a particular type of sensor, the books give me an idea of what constitutes professional grade tolerance and how these compare to what's available from the general robotics supply houses.

If you're a regular reader of this column, you know that I'm a fan of repurposing, especially when it comes to devices that were designed for uses unrelated to robotics. Take a few of my recent finds, discovered while I visited my local Guitar Center. The top find of the day was Moongel (RTOM Corp): gum-stick sized sheets of silicone gel designed to dampen vibrations of drum heads. One problem with drums is that they tend to resonate when nearby instruments — such as amplified bass guitars — hit their resonant frequency. The silicon gel adds mass to the paper-thin drum heads, thereby dampening the sympathetic vibration. I've also found that the blue gel sheets make excellent vibration-isolation platforms for miniature cameras and accelerometers.

Moongel is available at most musical instrument stores and online sites that sell percussion instruments. You can purchase a container of Moongel sheets on Amazon for about $7. Moongel is also sold in 7" and 14" diameter discs about 1/4" thick as practice pads for drummers. I've used the larger disc ($35) as a test platform for calibrating accelerometers. The pad is useful for isolating the sensors from vibrations conducted through my benchtop. It doesn't provide the isolation of a slab of marble floating on a bed of mercury, but for the price it suffices.

Another find at the music store was a set of drum tuners from Tama (**www.tama.com**) and Drum Dial (**www.drumdial.com**). These mechanical instruments — designed to directly measure drumhead tension — are an inexpensive ($60) means of checking the tension on fabric or Mylar covered wings, and control surfaces on R/C planes, boats, and hovercraft. I've used a drum tuner to calibrate the tension on Kevlar sheets laced with conductive thread connected to a Lilypad Arduino from SparkFun (**www.sparkfun.com**). Take a look at the drum dial in use on YouTube to see if it will work with your application.

My last find during my outing to the music store

was colored electric guitar strings by Aurora (**www.stringsbyaurora.com**). These coated steel strings are available in colors from black and white to gold, red, purple, and nitro lime ($15 for six cables). Why would you want these? Mainly for aesthetics. Why use a plain silver wire for your new claw gripper when you could have nitro tangerine? More practically, if you're demonstrating, say, a hand with 20 or more control cables articulating the joints, using a uniquely colored control cable for each servo or function can clarify operation and aid in troubleshooting. The main limitation of colored guitar strings is that they're not designed for heavy loads. If you need thick control cables, consider bass guitar strings from the same manufacturer.

The takeaway is that it pays to keep your eyes open for tools and technology that can be applied to your robotics projects. Otherwise, you're constrained by the economic decisions of the parts buyers for robotics supply houses. Have fun hunting. **SV**

FOR THE ROBOT INNOVATOR **SERVO** MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Jenn Eckert, Tom Carroll, David Geer, Dennis Clark, R. Steven Rainwater, Kevin Berry, Gordon McComb, Greg Intermaggio, Matt Bunting, John Blankenship, Samuel Mishal, Steven Nelson, Chris Olin, Mike Jeffries, Pete Smith, Alan Downing, Andrew Downing, Matthew Downing, Nathaniel Lewis, Frank Ou, Robert Ou

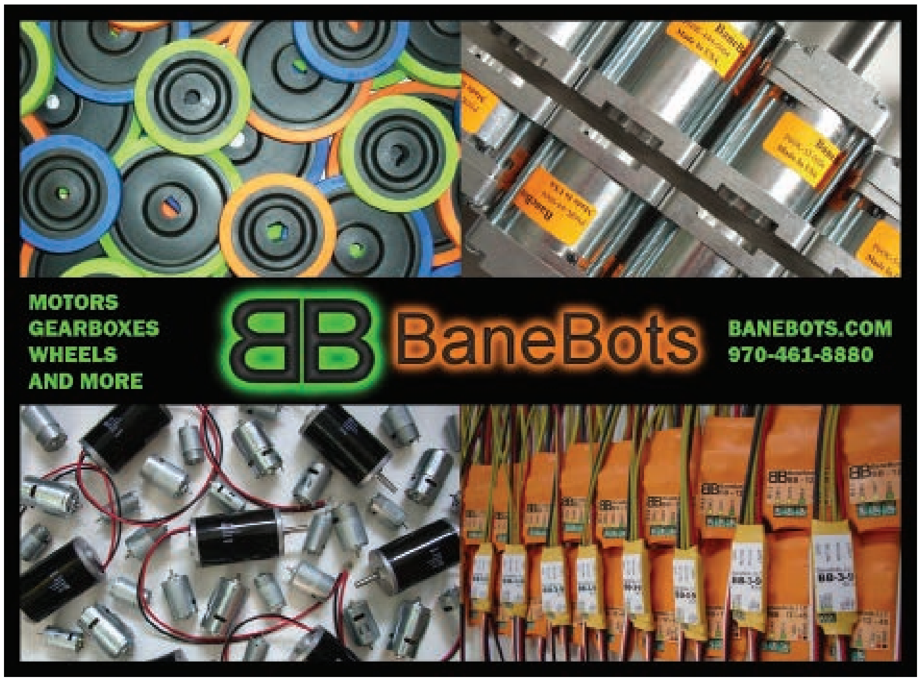
CIRCULATION DIRECTOR
Tracy Kerley
subscribe@servomagazine.com

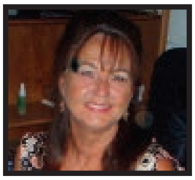
MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Shannon Christensen



Robytes

by Jeff and Jenn Eckert

Discuss this article in the *SERVO Magazine* forums at http://forum.servomagazine.com.

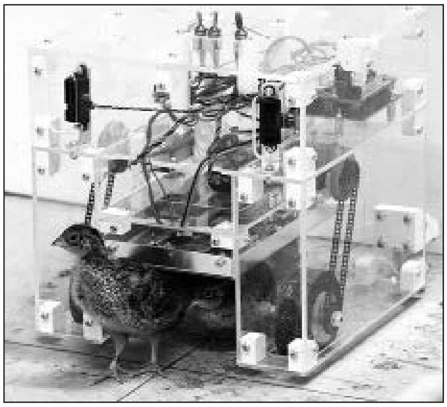
Bots At Center of Chinese Space Effort

Display model of Tiangong I module. Credit: Gregory Kulacki/People's Daily Online.

It's no secret that the Chinese intend to become a major player in the space race and, in fact, a display model of the Tiangong I (trans. Heavenly Palace) module was recently given public view at the China Academy of Space Technology (**www.cast.cn**). The lift date has been put on hold pending analysis of the August 19 failure to launch an experimental satellite — the result of a dud launch rocket. When it does enter orbit, however, this will constitute the first step toward completing a 60 ton space station by 2020.

"The ability to do that robotically is going to certainly be a technological step forward for them," said Joan Johnson-Freese, chairwoman of the Department of National Security Studies at the Naval War College in Newport, RI. "Some people have compared this to where we were at with Gemini. But we were doing it with people. If they can do it with robotics, it's a demonstration of a technological step forward."

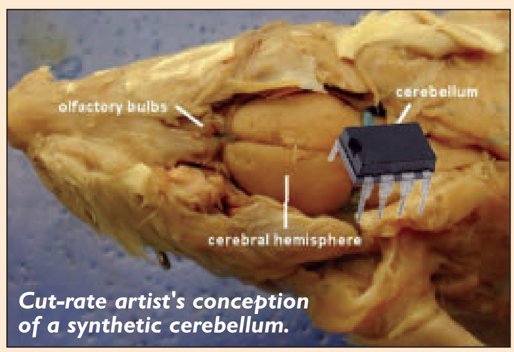
Quail vs. Robots: Questionable Research, but a Great Lunch



Seven-day-old quail chicks stand with the heated robot. Credit: E. de Margerie, et al. © 2011 IOP Publishing Ltd.

It's difficult to put the whole thing together, but we surmise that some researchers at the University of Rennes (**www.univ-rennes1.fr**) were discussing lunch one day and decided on caille en escabeche. While harvesting a covy of quail, academic curiosity got the better of them, and someone wondered if a robotic hen can raise chicks as well as a real one. Dying to find out, they divided 24 Japanese quail chicks into six groups of four. At the age of 36 hours, three groups were placed in a cage with a heated mobile robot and the other three in a cage with the same type of robot, only with locomotion disabled. For one hour per day for 10 days, the chicks were allowed to interact with the robots. From here on, it gets a little murky, but it seems that the chicks that spent time with the mobile robot temporarily navigated the cage faster than the others on their own. However, the others caught up after five days. It also was shown that the chicks that were exposed to the mobile robot emitted more distress calls, but that effect wore off after only one day. Somehow, the researchers found this significant and concluded that the study "shows that robots can underscore the results of past studies that show the importance of a mother hen's mobility on the normal development of her chicks." More importantly, they discovered that a bottle of Château Philippe-Le-Hardi Mercurey goes perfectly with seared quail, and will run you only about $22.

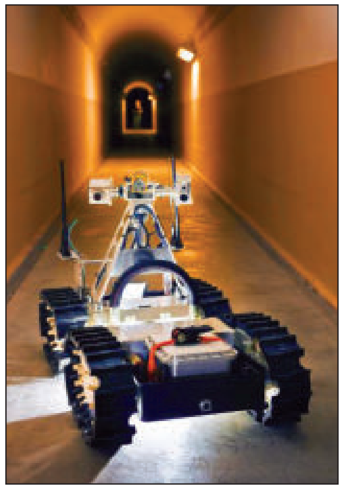
Rat Brain Enhancement



Cut-rate artist's conception of a synthetic cerebellum.

Another breakthrough of sorts has been announced by Matti Mintz — a professor of psychobiology at Tel Aviv University (**international.tau.ac.il**). Although more in the realm of cybernetics than strict robotics, his work does involve replacing living tissue with an electrical device, so we're letting it slip through. Much like cochlear implants and prosthetic devices can restore physical functions, Prof. Mintz's development of a synthetic cerebellum offers the promise of restoring brain functions in higher animals. In the case at hand, Mintz et al. analyzed brainstem signals feeding into a rat's cerebellum and noted the generated output. They then anesthetized the rat, disabled its cerebellum, and hooked up their synthetic version. At that point, the team taught the rat a conditioned motor reflex (i.e., a blink) by generating an auditory tone while hitting its eye with a puff of air. Eventually, the animal blinked on hearing the tone, even without the puff, thus proving that the chip was working. According to Mintz, prostheses based on this principle "might one day be used to enhance brain function in healthy people — to speed up learning or enhance memory." It was also noted that we may have to wait until the end of the century for replacements for specific, well-organized brain parts such as the hippocampus or the visual cortex, so you still need to avoid head injuries.

Remote-Controlled Mine Rescues



The Gemini-Scout is equipped to handle many obstacles — including rubble piles and flooded rooms — to help rescuers reach trapped miners.
Photo by Randy Montoya.

You might not know it from all the press coverage lately, but the number of US mining disasters has actually been dropping from a high of 356 in the period of 1901 to 1925 to only 17 from 1976 to the present, according to the US Mine Rescue Association. Nevertheless, we're talking about 60 to 70 mining accident deaths in the USA every year, and thousands around the world, so it looks like Sandia National Laboratories' (**www.sandia.gov**) Gemini-Scout Mine Rescue Robot is a pretty good idea.

Designed to scoot ahead of human rescuers, detect dangers, and provide relief to trapped miners, the bot can move through up to 18 inches of water, crawl over boulders and rubble piles, and even help plan rescue operations. Less than four feet long and two feet tall, the unit can navigate around tight corners and over safety hatches a foot high. It also can haul food, air packs, and medicine to trapped miners.

Gemini-Scout is equipped with two-way radios and can be configured to drag survivors to safety. The remote-controlled machine also sports gas sensors and a thermal camera to help locate survivors, plus a high-mounted pan-and-tilt camera that spots obstacles at a distance. Its electronics are housed in explosion-proof casings, so it doesn't have to worry much about methane. Unfortunately, the bot isn't expected to be commercially available until the end of next year, and the Mine Safety and Health Administration will be the primary customer for now.

According to Jon Salton, Sandia engineer and project manager, "We anticipate that this technology is broad enough to be appealing to other first responders, such as police, firefighters, and medical personnel. Gemini-Scout could easily be fitted to handle earthquake and fire scenarios, and we think this could provide real relief in currently inaccessible situations." **SV**

GEERHEAD

by David Geer

Contact the author at geercom@windstream.net

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

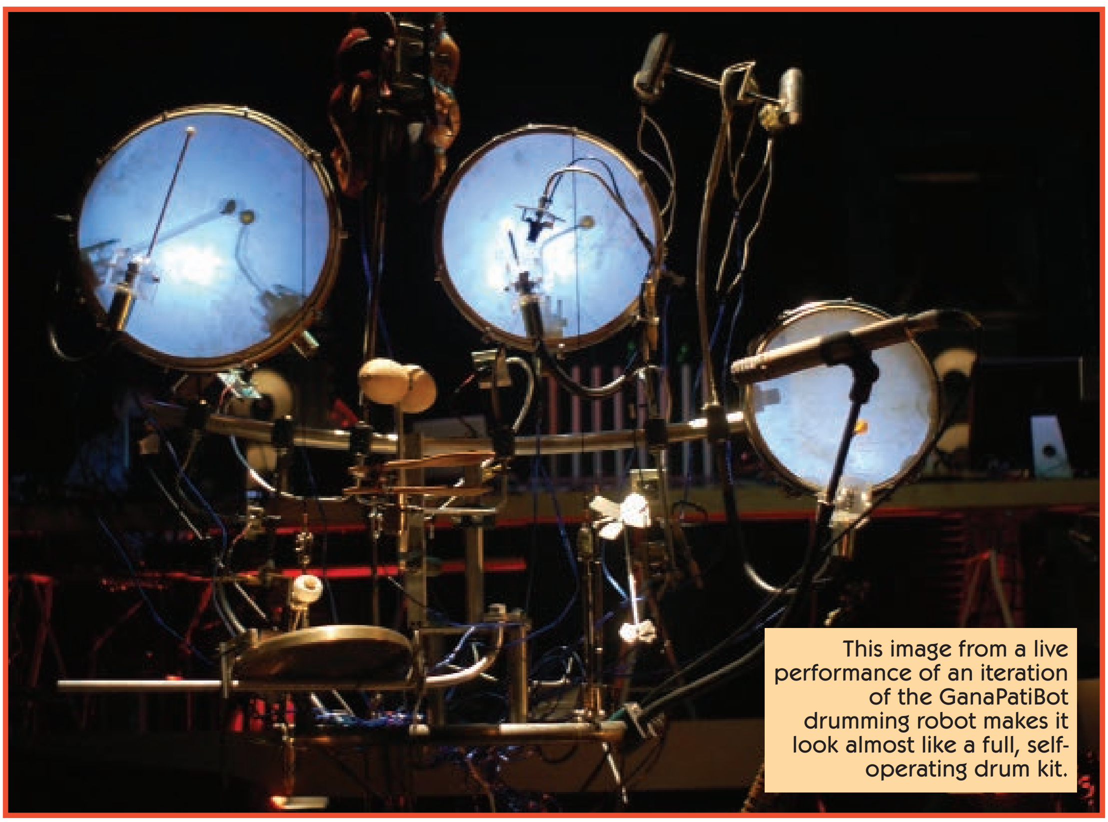
The KarmetiK Machine Orchestra

Some musical robots are meant to lead a band such as the conducting robots that appeared in this column called "Baton-Wielding Bots Command and Control Orchestrated Wonders!" (October '10). Others are meant to follow the lead of human conductors. Such is the case with the KarmetiK Machine Orchestra, which performs in the Valencia Art School's Walt Disney Modular Theater. It is an ensemble of Indian-derived robotic musical machinery made of real instruments, solenoids, and scraps from junkyards.

Under the co-direction of Ajay Kapur and Michael Darling, the Orchestra performs original music for a live audience with the aid of programming and the hands of the robotic instrument's creators who are students in the CalArts' Robotic Design for Music and Media class.

Let's meet the band, so to speak.

The GanaPati Robot



This image from a live performance of an iteration of the GanaPatiBot drumming robot makes it look almost like a full, self-operating drum kit.

The GanaPatiBot is a unique musical robot the class builds fairly regularly, having built and re-built one four times already from scratch. "One iteration of GanaPati had a budha head that moved up and down while the automatonic drum played," explains Ajay Kapur, Orchestra Leader and Instructor. One example of GanaPati used a disassembled sculpture and a star with drums on each point to produce its scintillating sounds.

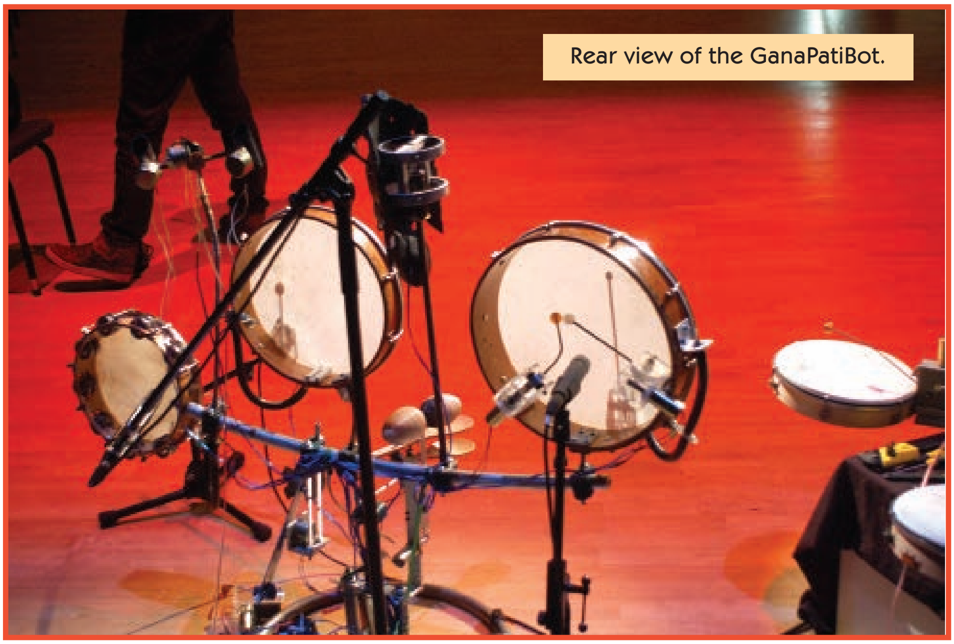
The GanaPatiBot was originally designed by both Kapur and Darling. The metal work was completed

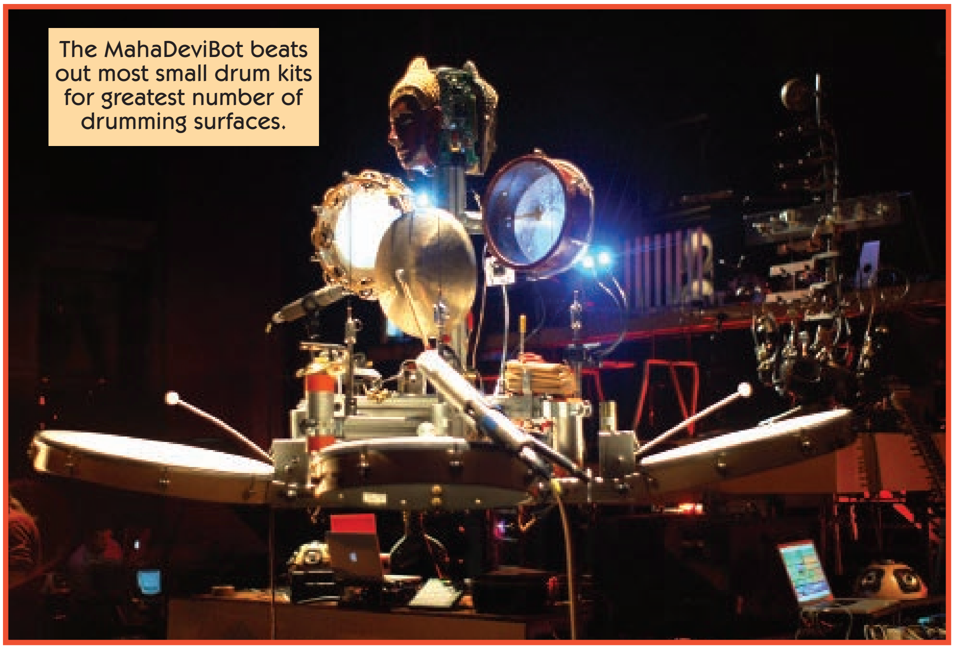
by student Matthew Setzer (for the 2008 version).

An upgrade from the MahaDevi robot of years past, the GanaPatiBot has multiple drums, each with its own solenoid systems for striking the drum which enables a variety of textures, as well as increasing the speed of a drum rool to well beyond any humanly achievable speed, Kapur commented.

One year, the roboticists put a propeller Leslie system at the rear of the GanaPatiBot with two speakers placed at either end of a motorized, spinning bar which played sounds and drones from an iPod Mini attached near the middle of the bar. Because the speakers flew off making the setup dangerous, the team no longer uses this configuration.

The students and instructors develop a new GanaPatiBot each year using their own style, aesthetics, and system of building. "We upgrade everything year after year, including the electronics," says Kapur.


Rear view of the GanaPatiBot.


The MahaDeviBot beats out most small drum kits for greatest number of drumming surfaces.

The MahaDeviBot

This was the first robot musician/instrument that Kapur built for his PhD project while on Vancouver Island. "It was the proof of concept that I could do all this work. I built it out of 80/20 slotted aluminum," says Kapur.

The MahaDevi uses a circuit board — the circuit was designed in CAD and manufactured especially for the class — to actuate each arm and corresponding solenoid. "I write the code for the students so they can begin to work with this robot," Kapur explains.

Kapur turned the robot into a MIDI device. "It takes a lot of programming, which in itself is an entire stage of the design and build process," explains Kapur. Still, the students are doing most of the actual performances on the robot, in real time. The robots can also be pre-programmed to perform pieces autonomously.

The robot's brains consist of Atmel chips and circuits, Arduino's open electronics prototyping, and a ChucK server to run the server that controls it all. The data transfers via MIDI over USB technology.

Designed by Kapur, the robot is a robotic extension of

Saving on Solenoids

The CalArts class that builds the mechanized musical wonders saves big bucks on solenoids by searching for older models in the discarded typewriters, printers, and other devices that line the local junkyards. "It is a savings and it is cool because we are recycling the solenoids," says Ajay Kapur, Orchestra Leader and Instructor, CalArts.

This multi-beater drumming bot looks as much like an old fishing reel as an instrument.

the North Indian musical ensembles, keeping time and creating complex rhythms. The robotic drum robot strikes a dozen different percussion instruments taken from various parts of India. The percussion instruments include frame drums, finger cymbals, bells, gongs, and wood blocks.

The NotomotoN1, the Tammy, and the GlockenBot

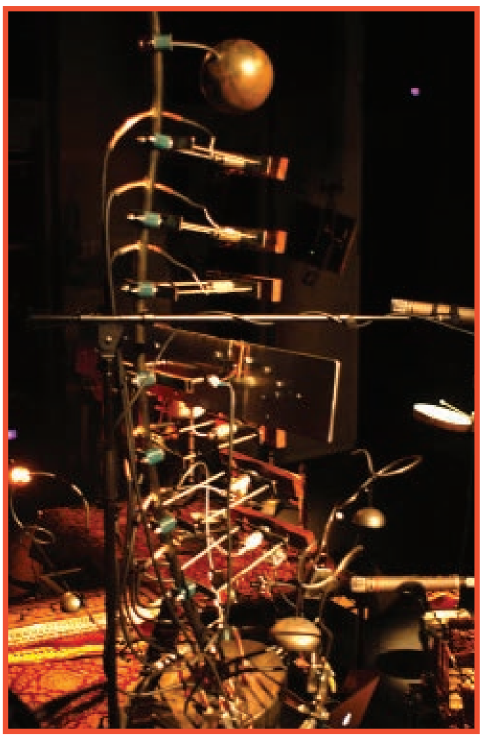
This one is a new entry into the aesthetically pleasing sound producing robot. "We started building it just last year. We created the drum's shell ourselves," says Kapur. The team wanted to create a drum that could beat very, very fast. To enable this, creators Kapur and Darling installed 12 "beaters;" six on each side of the drum, all beating at high speed.

"The robot has beaters that strike different parts of the drum for different timbres and if you get them going all at once, you can get high speed rolls that you could never get with a human being," Kapur explains.

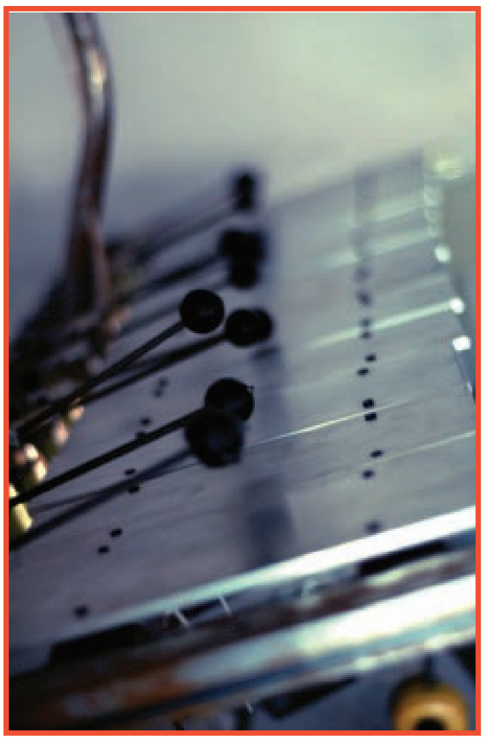
The team of Kapur and Darling built the Tammy in their very first edition of the Robotic Design for Music and Media class. "A couple of student groups were creating these marimba bars. We got all the parts from the scrapyard. It sits on a chair for mobility and uses telephone bells, marimbas made of rosewood for the chromatic scale, and a droning string from a fan to make music," Kapur details.

Together with Owen Vallis, Jordan Hochenbaum, Charlie Burgin, Jim Murphy, Jeff Lufkin, Steve Rusch, and Dimitri Diakopoulos, the robot's creators saw something in their minds that would embody many instruments, which would be added or developed for Tammy over the years. Tammy is six feet high and currently uses 14 actuators.

Made from a steel frame with aluminum bars that a CalArts Master's student cut to varying lengths to tune them musically to play the chromatic scale, the GlockenBot uses solenoids from junkyards and plays 13 notes, according to Kapur.



This looks like an instrument from a Dr. Seuss book.



The GlockenBot, which looks very similar to a xylophone.

Musical Robots United

"We network all the robots

Resources

KarmetiK Machine Orchestra Website
www.karmetik.com

Photos of the KarmetiK Machine Orchestra's Instruments and Performances.
http://calarts.edu/news/office-public-affairs/press-photos

Recent Coverage on KNBC TV:
www.latimes.com/la-ca-robot-orchestra-20110424,0,6582095.story

Los Angeles Times Print and On-line
www.nbclosangeles.com/news/local/Robots_Are_Taking_Over_The_World_Los_Angeles-117945114.html

Los Angeles Times Culture Monster
http://latimesblogs.latimes.com/culturemonster/2011/04/robots-are-ready-to-jam.html

Associated Press
www.evri.com/news/ap/tPvwE5SUlFlguGD6jy_Hu9QnhhtVuA

Associated Press Article with Video:
www.thestar.com/videozone/990828—robot-orchestra-jams-with-humans

Newsweek/Daily Beast
www.newsweek.com/2011/05/01/newsbeast-ideas.html

Wired
www.wired.com/underwire/2010/05/robot-orchestra

NPR
www.npr.org/templates/story/story.php?storyId=127140593

CNET
http://news.cnet.com/8301-17938_105-20004154-1.html

Create Digital Music
http://createdigitalmusic.com/2010/01/25/building-a-hybrid-man-machine-orchestra-pt-1-ajay-kapur-and-michael-darling

http://createdigitalmusic.com/2010/04/22/hybrid-man-machine-orchestra-interfaces-interaction-and-keeping-it-together

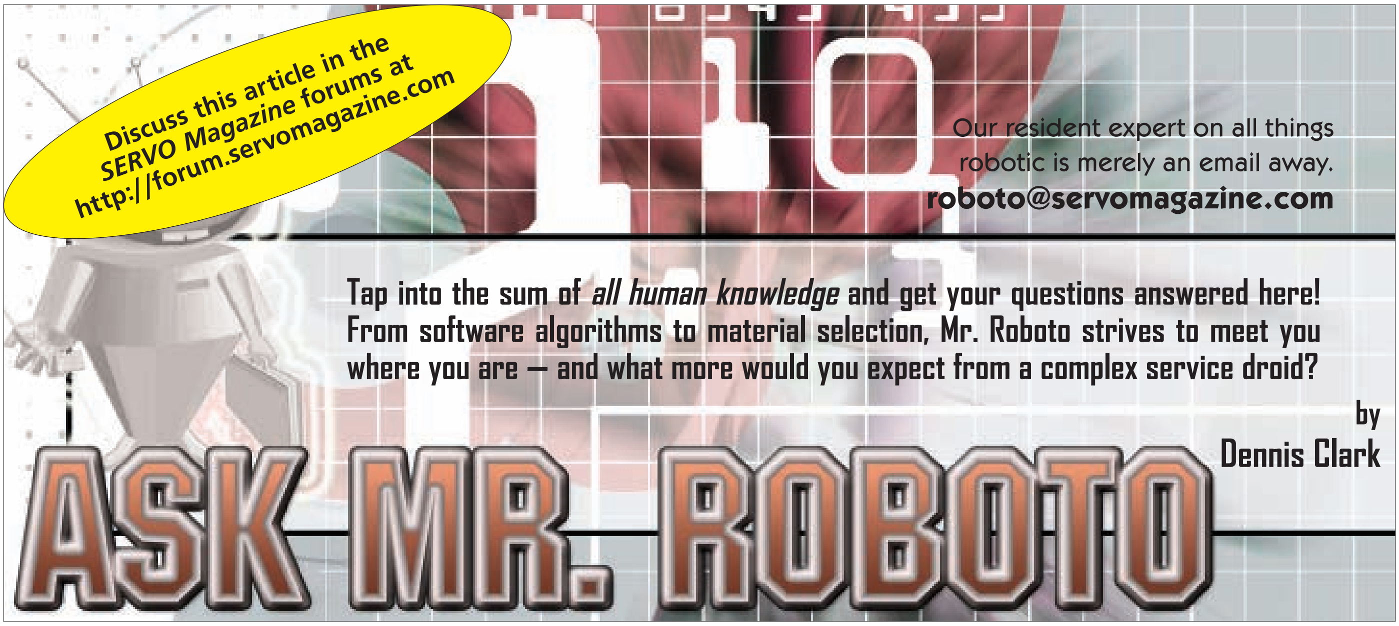
KarmetiK Video Trailer
http://vimeo.com/15245959

NotomotoN Video
www.youtube.com/watch?v=gZZXfldvojk

into one server in our main lab. This central server enables each student to connect wirelessly (or by wire) to the server and on to the robots to communicate and compose with their robots. Up to 20 students control the robots at the same time," says Kapur. Each student must learn programming to do this. They can write unique creative pieces, classical music, jazz, Indian, experimental, you name it.

Conclusion

"We have a great collaboration here. The theater students become involved in the building process because they work with theater sets, and my students learn about sensor design, electronics, and computer science," Kapur asserts, explaining the educational benefits of the class. "We sort of trick these artists into becoming engineers," Kapur comments. This gives students real working knowledge in multiple disciplines which will make them useful for a variety of career paths. SV



Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

Our resident expert on all things robotic is merely an email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

by Dennis Clark

ASK MR. ROBOTO

This month's column is dedicated to software issues that many of you are struggling with, attempting to go around, or are just plain avoiding by watching another edition of The Daily Show, rather than wrestling a problem to the ground and pinning it.

So, this time I'm going to answer these software questions that I've been queuing up for just such an occasion. I've chosen timers and Interrupt Service Routines (ISR) for my topics, using as many compilers as I can on the two most commonly used platforms I get questions about: the Atmel ATmega and PIC series chips. Contrary to popular belief, Mr. Roboto has not used every compiler or chip extensively, just some here and there. Rather than repeat every question and answer, I'll break the questions down into two categories and go over the nuts and bolts of how to solve the problems.

You will have to bear with me here. Working with timers and interrupts is rather like tying your shoelaces. Once you are "in the know," the process seems simple and is automatic, but explaining just exactly how and why one is doing each step is very laborious. So, I will start at "level 0" and work forward. I hope that you will then understand and be able to apply the process to your own project, and not just have to copy the code you find in these pages.

Q. So, let's get started with our first generically posed question. How do I set up a timer?

A. I'll start with the always popular ATmega168 or ATmega328 parts. These are commonly used on a variety of popular robot controllers like the Pololu Orangutan and the Arduino series. I use them pretty often myself. So, how do you set up a timer? Let's take a look. I promise, it won't hurt and after you do it a few times, it will be automatic to you.

The Compiler

I'm going to use the avr-gcc as my compiler of choice. It is a good compiler, runs on Windows, Mac OSX, and Linux, and it's free. (What more could you want?) You don't have that? Well, go back through your *SERVO Magazine* archives and find where I detail how to set it up, either using *Winavr* or *Eclipse* (my choice) in the August '08 article.

Homework

I am not going to reproduce the entire 376 page document Atmel provides for us to configure and use the ATmega168/328 parts. You should get a copy of it from your favorite parts supplier or Atmel (**www.Atmel.com**) to use as a reference while you read this article. Another good place to get the ATmega168 datasheet is your parts distributor; it will be easier to find there. I use Digi-Key (**www.digikey.com**). Simply search for the ATmega168 and click on the PDF symbol where the part is described.

The Timers

There are three timers in the ATmega168/328 parts: TMR0 and TMR2 (eight-bit), and TMR2 (16-bit). All of these timers can do PWM, or they can be configured to just be timers. TMR2 can also be configured as an *Input Capture*, but that is another story. I'll look at TMR2 for this discussion.

While all of the timers can take either an I/O pin input or use the system clock to advance the timer, TMR2 can also use an external crystal, so this one can be used with a 32.768 KHz source to run a real time clock. We're not going there this time. We'll select the system clock, which in the case of an Arduino is typically 16 MHz. So, I'll just use 16 MHz as our clock source. If you use another system frequency, substitute that frequency in when doing the timing calculations that I'll show you.

The Timer Registers

To set a timer, you need to fiddle with several registers. The following registers set up the clock source, pre-scaling, and what the timer is used for. Each timer will have its own unique set of registers. They will all use this naming convention; just change the number to match the timer/counter number.

TCCR2A: Timer/Counter Control Register A. Configures PWM configuration on OC2A/OC2B pins, if we use them for PWM.

TCCR2B: Timer/Counter Control Register B. Configures Timer 2 pre-scale settings and a little more of the PWM settings, if used.

TCNT2: Timer/Counter Register. This holds the current timer/counter value.

ASSR: Asynchronous Status Register. Allows selection of external clock sources.

These next registers are used to match PWM periods — more on that later; they aren't needed if you are just using the timer.

OCR2A: Output Compare Register 2A. Match value for PWM on OC2A pin.

OCR2B: Output Compare Register 2B. Match value for PWM on OC2B pin.

This next set of registers allow us to set interrupts on the timer conditions and check for interrupt flags.

TIMSK2: Timer/Counter Interrupt Mask Register. Configures the type of interrupt (if any) you want to have on PWM or Timer conditions.

TIFR2: Timer/Counter Interrupt Flag Register. We can look here for flags set by configured interrupts.

Setting Up a Timer

We are going to start by just setting up a timer — not PWM — so these settings will reflect that. **Listing 1** shows the init routine. I'll comment on what was done.

1). Always clear your flags before you turn something on, especially interrupts.

2). Have the timer match on the OC2A setting. Table 17-8, mode 2, CTC is the counter/timer clock match on OCRA. When this match occurs, the clock is cleared back to zero to start over.

3). Use the internal clock with no prescale. Table 17-9 shows the prescale settings. Use other settings if you want a slower clock.

4). Clear the timer to zero; start from a known place.

5). Here we set the interrupt. A timer just running by itself is useful for some things; don't use interrupts if you don't need them. Here we will use the timer match to fire an interrupt (more on that later). I've shown two ways to set a bit in a register.

6). 160 = 10 µs (microseconds) is the timer match with a 16 MHz system clock. I did this because it was a good value to use for controlling an RC servo pulse.

Listing 1: Setting Up the Timer.

```
TIFR2 = 0;                                          // (#1)
TCCR2A = 0x02;                                      // (#2)
TCCR2B = 0x01;                                      // (#3)
TCNT2  = 0;                                         // (#4)
TIMSK2 |= (unsigned char)_BV(OCIE2A);               // (#5)
TIMSK2 |= (unsigned char)(1 <<OCIE2A);              // (#5)
OCR2A = 160;                                        // (#6)
```

By now you're thinking, "Why use an interrupt?" There are good reasons for your timer to cause an interrupt. One is to provide a background *ticker* for timing your program. I find that such a timer is handy for timing state machines and checking for timeouts in various places in the program. Every time the interrupt goes off, you can increment a large counter in your program. In this way, you can check timing for many things without blocking program flow. **Listing 2** shows what the ISR for just such a timer might look like.

My ISR does two things: it controls the position of an RC servo (match count), and it handles my system timer tic (1_1ms) at a 1 ms resolution. Handy, isn't it? If all you

Listing 2: Timer ISR.

```
ISR(TIMER2_COMPA_vect)
/*
 * 10 microsecond ISR on TIMER2, a 8 bit clock
 */
{
        static uint8_t       t_10us;
        static uint16_t      match;          // The single servo used
        static uint8_t       next_pulse;

t_10us++;
        if (t_10us == 100)                   // 1ms background tic
        {
                t_1ms++;
                t_10us = 0;
        }

        match++;                             // increment every 1
        if (match > next_pulse)
                SPIN = 0;                    // drop servo bit
        else
                SPIN = 1;                    // raise servo pin high

        if (match > 1900)
        {
                next_pulse = servo;
                match = 0;                   // restart servo timer
        }
}
```

Listing 3: Setting Up PWM.

```
void InitMotors(void)
/*
 * Set up the motors and PWM and such.
 */
{
        DDRD |= (unsigned char)(1 <<PD5);  // (#1)
        DDRD |= (unsigned char)(1 <<PD6);

        TCCR0A = 0xF3;                     // (#2)
        TCCR0B = 0x03;                     // (#3)

        OCR0A = 0;                         // (#4)
        OCR0B = 0;
}
```

needed to do was keep a 1 ms background tic, you could set up the timer to interrupt at a 1 ms rate by choosing a pre-scale value and match count that would cause the interrupt every 1 ms. (Hint: 16 MHz/128 * 125 = 1 ms).

That wasn't so hard, was it? The most difficult part of this process is probably finding out how avr-gcc wants you do set up interrupts (which I also showed you). But how did I find the name of the interrupt so it would call the correct ISR? Okay, you got me. That was *really* hard to find, so I went to good ol' Google, and it found **www.nongnu.org/avr-libc/user-manual/group__avr__interrupts.html** which set me right up. Somewhere in the avr-gcc documentation it says this. I was sure, but I didn't find it without Google! At least now *you* don't have to search for it.

Using a Timer for PWM

Our first step has nothing to do with timers at all! We're going to turn off the *comparator module* on the chip. This module deals with comparing analog voltages and causing things to happen by comparing them. This module defaults to on and will cause you no end of headaches with your I/O if you don't disable it!

```
ACSR = 0x80;         // turn off the comparator
```

This time, I'll pick TMR0 for our PWM. In this series of microcontrollers, a PWM block typically controls two PWM output pins. In this case, our PWM outputs will be called OC0A and OC0B, for *Output Compare Zero A* and *Output Compare Zero B*. We'll use similarly named registers in a different way. In fact, we could just as easily have used TMR2 or even TMR1 for PWM, but I thought that I'd change it up a little. Yes, this is just an eight-bit PWM, but for most of us, we use about three speeds: off, slow, and fast. Eight bits does that just fine.

Timer0 PWM Registers

TCCR0A: Timer/Counter Control Register A. Configures PWM configuration on OC0A/OC0B pins, if we use them for PWM.

TCCR0B: Timer/Counter Control Register B. Configures Timer 0 pre-scale settings and a little more of the PWM settings, if used.

OCR0A: Output Compare Register 2A. Match value for PWM on OC2A pin.

OCR0B: Output Compare Register 2B. Match value for PWM on OC2B pin.

DDRD: Data Direction Register PORTD. Set the direction of data transfer for the pins on port B. In this case, OC0A and OC0B are PD6 and PD5, respectively.

Setting Up the PWM

Above we disabled the comparator module. This assures us that our I/O pins are really digital. Now, we'll configure the registers above to PWM some motors. Since we're going to send a PWM signal out, we'll need to deal with the data direction registers, as well as the timer registers. **Listing 3** shows how to initialize all of the needed registers.

Set the DDRD register for outputs on PD5 (OC0B) and PD6 (OC0A). Set the fast PWM mode, clear the output on match, count up from the bottom (0). Fast PWM will only count up from 0, match (clear output), count the rest of the period, then restart at 0 again with the output set to 1. Set the prescale to 64 which on a 16 MHz part means the clock will be 250 kHz divided by 256 (an eight-bit count) which translates to a PWM rate of 977 Hz. Set both PWMs to 0. We should always start out stopped, don't you think?

At this point, all you need to set a motor speed is to put a non-zero value into OCR0A or OCR0B, and the motor will go that speed. This is *simple* motor speed control! If I was going to use a PID algorithm, I would use the PWMs on TMR1 which is 16 bits of resolution; this would give me a smoother PID loop with more control. For simple motor control, eight bits is plenty.

Wrap-Up for the ATmega168/328

Using these techniques, you can build your next robot with a background timer that you don't need to pay attention to — except to read it. I like to use a 32-bit variable and either a 1 ms or 10 ms tic; these will last a good long time. (Get out your calculators class! There will be a quiz: How long will it take for a 32-bit counter to turn over to zero when incremented every one millisecond?) You should do some defensive programming, however, to make sure you don't roll over to zero between when you take a time reading and wait for a certain period of time to pass. When dealing with PWM, it is even simpler than setting up a timer with an ISR. Now, go do it!

Next Time, PIC Microcontrollers

I'll be writing about a few different PIC micros and a couple of different compilers next month. PICs are different than the ATmega parts, but not more difficult to use. Well, we've come to the end of another Mr. Roboto! Keep those letters rolling in to roboto@servomagazine.com. SV

EVENTS Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**

— **R. Steven Rainwater**

DECEMBER

1-4 **ROBOEXOTICA**
Vienna, Austria
Watch robots mixing cocktails, lighting cigarettes, and working toward other ground-breaking achievements in electronic cocktail culture.
www.roboexotica.org

10 **Robotic Arena**
Wroclaw, Poland
Autonomous robots compete in Sumo, mini Sumo, line following, and freestyle events.
http://lirec.ict.pwr.wroc.pl/~arena

15-18 **IROC International Robot Olympiad**
Jakarta, Indonesia
This year's theme for the international student robot contest is "Robots helping people from natural disaster."
www.iroc.org

27-29 **MindSpark**
College of Engineering, Pune, India
Events include Micromouse, Dogfight, and Search and Destroy.
www.mind-spark.org

JANUARY 2012

6-8 **TECHFEST**
Indian Institute of Technology, Bombay, India
This year's contests for autonomous robots include ArchiTech, Autobots, Nexus, Robowars, and Split Second.
www.techfest.org

13-14 **FIRST LEGO League of Central Europe**
Brandenburg University of Technology Cottbus, Germany
The Central Europe regional for the FIRST LEGO League — a robot competition for students ages 10 to 16.
www.hands-on-technology.de/en/firstlegoleague

24 **Powered by Sun**
Ostrava, Czech Republic
A race of solar-powered autonomous robots.
http://napajenisluncem.vsb.cz

26-29 **ION Autonomous Snowplow Competition**
St. Paul, MN
The folks who run the annual autonomous lawnmower competition bring you this new cold-weather robot event for autonomous snowplow robots.
www.autosnowplow.com

27-30 **Robotix**
IIT Khargpur, West Bengal, India
Lots of events for autonomous and remote-control bots including Ballista, Pirate Bay, RAFT, RoboCop, Fugitives, and Negotiators.
www.robotix.in

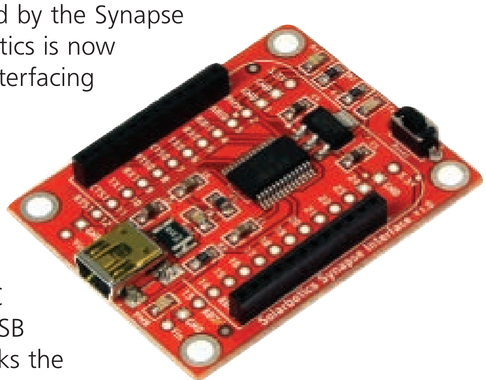
27-30 **Techkriti RoboGames**
Indian Institute of Technology, Kanpur, Uttar Pradesh, India
This year's events are Wild Soccer II, The Isle of Torgua, Robot's Got Talent, Lumos, and Ocean's Fourteen.
www.techkriti.org/#/competitions/robogames

Robotics Showcase

NEW PRODUCTS

Synapse and BrutusBot Kits

Familiar with XBee? Of course! How about Synapse — not so much? The geeks at Solarbotics have been playing with the new Synapse 802.15.4 modules and feel they're more handy than the venerable XBee. Since Synapse modules offer more I/O, embedded microcontroller Python, A2D converters, and I²C, they offer a great way to upgrade wireless projects, especially with communication ranges up to 1,000 ft indoors and three miles outdoors (line-of-sight). To easily harness the new features offered by the Synapse RF100, Solarbotics is now offering two interfacing kits.



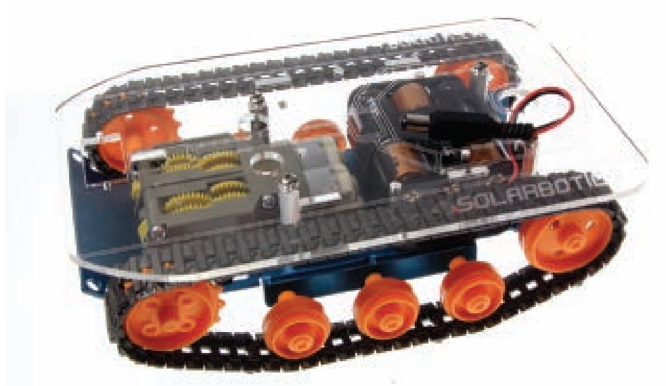
The Synapse-to-FTDI adapter kit interfaces the Synapse module to a PC with an FTDI USB converter, breaks the pins spacing out to 0.1", and generates the 3.3V power necessary to drive the module. Price is $22.95 USD.

The Synapse-to-FTDI kit features:

- Fuse-protected USB-TTL interface (FT232RL chip).
- 3.3V 1A onboard voltage regulator.
- Tx/Rx/Power LEDs.
- Breadboard-compatible breakout headers.
- Reset switch (handy when erasing SNAPpyscripts or upgrading firmware).
- #4-40 mounting holes.

The Synapse-to-Xbee Adapter Kit converts a Synapse module to a drop-in replacement for Xbee RF nodes. Since the Synapse has extra I/O the XBee doesn't (and with embedded python), the adapter breaks out these points to LEDs and header pins for increased hacking power. Users can convert Arduino XBee projects to Synapse with this converter. Price is $7.50 USD.

Solarbotics also introduces a new, more useful platform based on the Tamiya Bulldozer technology called the BrutusBot Tank Platform.



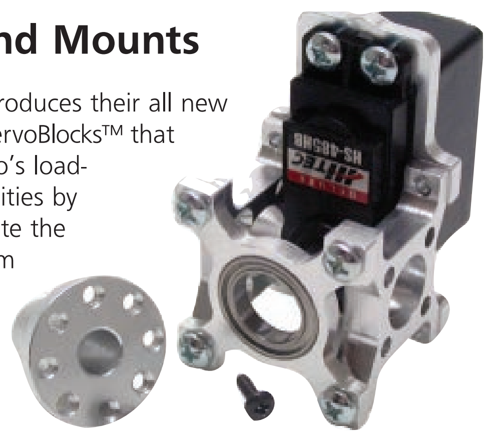
The BrutusBot features the Tamiya Gearbox and Tread set, anodized aluminum frame, and Arduino-compatible acrylic top-plate. Being just a motorized base, users can add any controller platform desired, from something as simple as an analog modified "Herbie" brain to as complex as a wireless Synapse networking module. Price is $52 and features include:

- Rugged anodized aluminum chassis with skid plate.
- Grippy Tamiya rubber treads for excellent traction.
- Tamiya twin motor gearbox configurable for high speed (indoor/smooth environment) or high torque (outdoor/rough environments).
- Acrylic top-plate with cutouts for pass-through wiring ports and mounting a sensor-sweeping servo.
- Interfaces with Solarbotics S.A.F.E. enclosures to protect electronics in off-workbench environments.
- Six-cell 'AA' battery pack for excellent motor power and lifetime.

For further information, please contact:

Solarbotics, Ltd.
3440D 11A Street NE, Suite #101
Calgary, AB T2E 6M6, Canada
866•276•2687 Fax: **403•226•3741**
Website: **www.solarbotics.com**

Blocks and Mounts



ServoCity introduces their all new patented ServoBlocks™ that increase a servo's load-bearing capabilities by helping to isolate the lateral load from the servo spline and case. The versatility of ServoBlocks

allow users to create complex, extremely rigid structures with ease using standard Hitec servos. The 1/2" aluminum hub shaft provides multiple mounting options using 6-32 screws. The robust 6061 T-6 aluminum framework acts as a servo exoskeleton, enhancing the mechanical loads the servo can withstand. ServoCity's new .770" hub pattern is repeated throughout the framework to allow endless attachment options. ServoBlocks are compatible with all standard size Hitec servos. The kit comes unassembled; servo not included. The five piece kit is $24.99.

Brand new and exclusive to ServoCity, these new vertical aluminum mounts work for nearly any servo mounting application. Constructed entirely from 6061 T6 aluminum, the mounts are durable and solid. The "A" frame design provides additional strength, while keeping a low profile to allow for mounting in tight spaces. The mounting kit includes two mounting brackets and four 6-32 x 1/4" pan head Phillips screws. These mounts are designed for use with standard size Hitec and Futaba servos. A single servo mounting kit is $6.99.

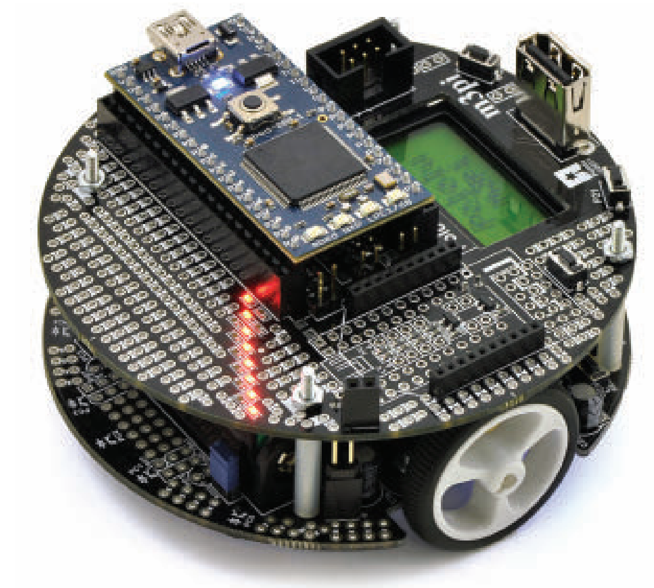
For further information, please contact:

ServoCity
620 • 221 • 0123
Email: marketing@servocity.com
Website: **www.servocity.com**

Pololu m3pi Robot

Pololu announces the release of the m3pi robot — a high-performance, expandable mobile platform for use with ARM's powerful mbed development board. When socketed in the m3pi, the 32-bit mbed makes an extremely capable high-level robot controller that can be easily interfaced with wireless serial modules, additional sensors, and custom electronics.

The m3pi robot consists of a 3pi robot base connected to an m3pi expansion board which includes sockets for an mbed development board and Wixel, or XBee wireless serial modules. The base ships pre-programmed with a serial slave program, so any microcontroller board capable of sending serial commands can be used as the m3pi's high level controller. Or, the 3pi base can be programmed directly. Since the heart of the m3pi is a 3pi robot, the m3pi has all the features of the 3pi robot, including a maximum speed of around 1 m/s, regulated motor power that prevents battery voltage from affecting performance, five reflectance sensors for line following and maze solving, an 8x2 character LCD, and a piezo buzzer for simple sounds and music.



A fully assembled m3pi robot with 3pi base included, is available for $149.95. An m3pi expansion kit, which can be used to convert a 3pi robot into an m3pi robot, is available for $27.95. An mbed module is not included.

For further information, please contact:

Pololu Corporation
Website: **www.pololu.com**

Upgraded EASYPIC v7

MikroElektronika announces the successor to their major development board for PIC microcontrollers: the EasyPIC v7. For the first time in EasyPIC's almost 10 year history, EasyPIC v7 has grouped PORT headers, LEDs, and buttons into input-output groups, thus making them easier to use. The boards are equipped with tri-state DIP switches, so placing pull-up or pull-down jumpers to desired pins is now just a matter of pushing the switch.

Connectivity is the main focus of EasyPIC v7 which provides three separate PORT headers in the input-output groups, and another one on the opposite side of the board. This way, users can access those pins from any side.

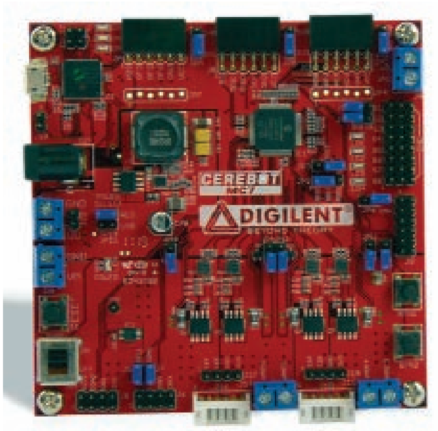
The board has a dual power supply, supporting both 3.3V and 5V microcontrollers. Another feature of the board is its powerful on-board mikroProg programmer and In-Circuit debugger, capable of programming over 250 PIC microcontrollers. Debugging is supported with all mikroElektronika PIC compilers (mikroC, mikroBasic, and mikroPascal). The board has three displays: GLCD 128x64, LCD 2x16 character, and four-digit seven-segment displays.

EasyPIC v7 is the first board that supports the mikroBUS pinout standard. It now has new modules which include: serial EEPROM, piezo buzzer, and support for both DS1820 and LM35 temperature sensors.

For further information, please contact:

mikroElektronika
Website: **www.mikroe.com**

CereBot MC7 Development Kit



Microchip Technology Inc., announces the availability of a Microchip dsPIC33 Digital Signal Controller (DSC)-based development kit. The Digilent Cerebot™ MC7 development kit addresses the growing interest in embedded motor control from the academic and hobbyist markets, and is ideal for learning about microcontrollers and solving real problems. The kit includes a demonstration board that provides four half-bridge circuits, eight RC servo motor connectors, the ability to use Digilent Pmod™ peripheral modules, and an integrated programming/debugging circuit that is compatible with the free MPLABIDE. Example applications include university embedded-systems and communications classes, senior capstone projects, and numerous other academic and hobbyist projects.

The Cerebot MC7 board features four half-bridge circuits that are rated for 24V at up to 5A. These half bridges can be used to control two brushed DC motors, two bi-polar stepper motors, one brushless DC motor, and one uni-polar stepper motor. An onboard 5V, 4A switching regulator with an input voltage up to 24V simplifies operation of the board, enabling it to operate from a single power supply in embedded applications such as robotics. The onboard dsPIC33 DSC features 128 KB internal Flash program memory and 16 KB internal SRAM, as well as numerous on-chip peripherals, including an advanced eight-channel motor-control PWM unit, an enhanced CAN controller, two Serial Peripheral Interfaces (SPIs), timer/counters, serial-interface controllers, an Analog-to-Digital Converter (ADC), and more. The Cerebot MC7 board combines two pushbuttons and four LEDs for user I/O, as well as connections for two I^{2}C busses — one of which contains an integrated serial EEPROM device."The Cerebot MC7 board is an ideal embedded motor control and general-purpose microcontroller experimentation platform for academics and hobbyists," stated Clint Cole, president of Digilent, Inc.

The Cerebot MC7 development kit is available for $119. It can be purchased from Digilent or from microchipDIRECT.

For further information, please contact:

Digilent	Website: **www.microchip.com/get/3N9U**
microchipDIRECT	Website: **www.microchip.com/get/DUMT**

5 MHz Function Generator With Frequency Counter

Global Specialties, has re-introduced the new 5 MHz function generator. The model 4005 has a number of features and functions not offered by other similar 5 MHz function generators on the market today.

The 4005 is designed to be very vestal and can be used as a function generator, sweep generator, pulse generator, and a frequency counter. Cost is $320.

VCF (voltage controlled frequency) produces precision sine, square, and triangle waves over 0.05 Hz to 5 MHz for sub-audible, audio, ultrasonic, and RF applications. A continuously variable DC offset allows the output to be injected directly into circuits at the correct bias level.

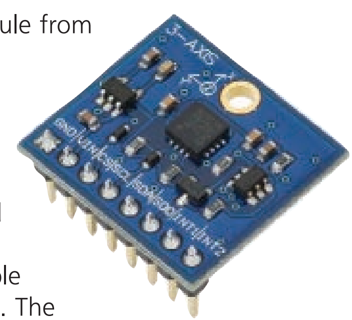
Variable symmetry of the output waveforms converts the model 4005 into a pulse generator capable of generating rectangular waves or pulses, ramp or saw tooth waves, and skewed sine waves of variable duty cycle. The sweep generator offers a linear sweep with a variable sweep rate and sweep width up to a 100:1 frequency change. The frequency response of any active or passive device up to 5 MHz can then be determined.

The 4005 can be used in applications in both analog and digital electronics such as engineering, manufacturing, servicing, education, and hobbyist.

For further information, please contact:

Global Specialties, LLC	Website: **www.globalspecialties.com**

Three-Axis Gyroscope Module



The new gyroscope module from Parallax is a low power, three-axis angular rate sensor featuring temperature data as an added bonus. Raw measured angular rate and temperature data are accessed from the selectable digital interface (I^{2}C or SPI). The module features a small package design and has an easy to access SIP interface with a mounting hole for quick connectivity to projects. It's designed for use with a large variety of microcontrollers with different voltage requirements. Price is $29.99.

For further information, please contact:

Parallax	Website: **www.parallax.com**


DF RobotShop
Rover
Compatible with
ARDUINO
MYROBOTS.COM
Connect your Robots. Reap the benefits™
Affordable, Flexible and Scalable
Compatible with thousands of off-the-shelf parts!
XBee
Wireless Control
Speech Control*
Mecanum
RobotShop
Robotics at your service!™
www.robotshop.com
Scan on your smart-phone with a bar reader app
ADDITIONAL VARIANTS TO COME
THE EVOLUTION CONTINUES...
RobotShop and MyRobots logos copyright RobotShop Distribution inc. Robotics at your service and Connect your Robots. Reap the Benefits are trademarks of RobotShop Distribution inc. DFRobot logo copyright DFrobot. Arduino logo copyright Arduino. Xbee logo copyright Digi International inc. *Headset and female operator not included with speech controlled model, which is provided with an integrated electret microphone.

bots IN BRIEF

DRAGON IT OUT

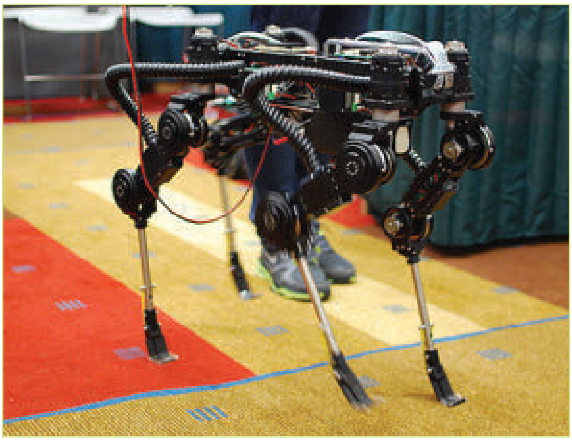
Dragons are typically giant, dangerous, and potentially scary. But dragons can also be cute and fuzzy and cuddly. Researchers at Northeastern University, Harvard, and MIT have gotten together and invented a little robot dragon designed to appeal to preschoolers. Fans of celebrity roboticists might recognize MIT's Cynthia Breazeal in the photo (to the left) on the far left; also pictured are David DeSteno from Northeastern (right), and Paul Harris from Harvard (far right).

The robot they're all fawning over is — believe it or not — a descendant of Nexi — MIT Media Lab's small humanoid. As you can see, it's a robotic dragon, called "dragon robot." The relation to Nexi comes in the form of research by Northeastern's Social Emotions Group, showing that things like eyes and movements have a very significant impact on how people relate to robots — especially when it comes to trust and communication in learning environments. DeSteno, an associate professor of psychology at Northeastern, explains:

"Certain non-verbal cues like mimicking behavior to improve rapport and social bonding, or changes in gaze direction to guide shared attention, are central. When kids learn from human teachers, these cues enhance the learning. We're designing our new dragon robots to be able to have these capabilities."

Specifically, the dragon robot is designed to teach preschoolers language skills. It's furry, extremely emotive, and the intention is that kids will be able to develop an emotional connection with it. Once they trust the robot like they would something that's actually alive, it'll be a much more effective teacher.

At this stage, the dragon robots are going to undergo some preliminary testing with preschoolers at MIT. Once the researchers figure out what social cues are the most crucial to developing those emotional bonds, the robots will venture out into the world as distance-learning tools to help kids in rural areas learn their shapes, colors, numbers, and fantasy animals.



SIMple QUADRUPED

This feisty little guy is a quadruped robot called SQ1. It's a project by the South Korean company, SimLab. Their RoboticsLab simulation software is being used to figure out how to get the quadruped to walk without actually having to risk a trial-and-error approach on a real robot. And it works!

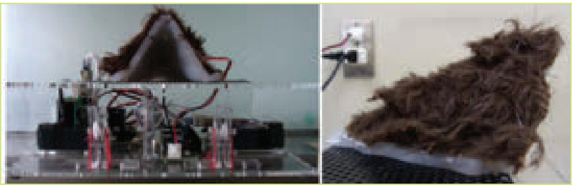
This smaller robot is being used to test out different gaits that have proven themselves in simulation, before the full-sized (and more expensive) version tries not to fall over.

bots IN BRIEF

BOT BIKER

The current generation of bicycle-riding robots is extremely complicated, relying on giant gyroscopes and thick wheels to keep themselves upright even while stationary. This is certainly a neat trick, but it's not something that most humans can pull off. It's not a problem that robots are better at something than we are, but there's something to be said for human emulation, too.

It turns out that getting a robot to ride a bicycle doesn't need to involve much more than a hobby level humanoid employing a relatively simple gyroscope that sends steering commands to keep things generally upright. This KHR3HV bipedal robot (which can be yours for about $2,200) has a nifty custom bike that it got from somewhere, and it can zip around under remote control at up to 10 kph, even making its own starts and stops.



CAT GOT YOUR EAR?

Robots have ears. They're called microphones and you usually find them just inside some tiny little hole somewhere. However, there are good reasons why certain animals have large ears. Namely, big ears allow animals to hear quieter sounds, and localize those sounds more precisely.

This is the idea behind "active soft pinnae" which is fancy roboticist speak for "ears that wiggle." The robotic ear in the picture is a reasonably accurate reproduction of a kitty ear — including a fake fur covering on the back and the ability to both rotate side to side and deform downwards. There's a microphone buried down inside the ear, of course, but the external structure is the important part.

So, what good is it? Testing has shown that it's possible to pinpoint the direction (azimuth and elevation) to a sound with just two wiggly ears instead of needing a complex microphone array. Furthermore, the ears can be used to localize sounds by moving independently of the head or body of a robot which is a much more efficient approach. Of course, ears like these are awfully cute, and with the addition of some touch sensors, you could give your robot that friendly scratching it deserves.

ROCK 'EM SOCK 'EM ALREADY

That didn't take long ... There is already an authorized Real Steel WRB Built For Battle set. It includes a 12 x 12" ring, Atom, Zeus, light up controls, and sound effects from the film. It's recommended for those over six years old. A Rock 'em Sock 'em by any other name ...

STICKY PROPOSITION

Robots use all sorts of clever techniques to climb. They use magnets, grippers, gecko feet, electrostatics, and even supersonic jets of air. It's sort of surprising, then, that the idea of using the most stereotypically sticky thing in the universe to climb has been (more or less) ignored until now. Yes, this robot sticks to surfaces with glue.

Technically, what this robot uses is hot-melt adhesive, or HMA. This is the stuff that comes out of hot glue guns, and it goes from a solid to a sticky liquid when it's passed through a heating element. As it cools, it solidifies again. The robot uses this property to temporarily bond its limbs to a vertical surface one by one and hoist itself up, unsticking itself as it goes by re-heating the blobs of glue that it sets down.

You've probably already thought of several issues that this robot has to deal with. First, it's very, very slow since it has to wait for the adhesive to cure every time it takes a step — a 90 second process. Second, it leaves a trail of sticky little glue spots along every surface that it climbs, making its usefulness questionable in many environments. So yes, a few things need to be addressed, but this technique has a bunch of upsides, too. The biggest one is that glue — being glue — sticks to just about anything. It doesn't have to be especially rough, especially smooth, or especially magnetic, which makes it more versatile than the current generation of just about every other robot adhesion system you can think of.

Also, the hot melt adhesive can support a lot of weight, and it can do it completely passively. You don't need to expend energy once the adhesive sets to keep from falling. The bonding strength of the HMA in its solid state is such that a four square centimeter little patch can hold a staggering 60 kilograms — easily enough to hold this robot plus a fairly gigantic payload, most of which is likely going to have to consist of extra sticks of glue.

FOAMING ROBOTS

Robots are quite good at doing very specific tasks. Arguably, doing very specific tasks are what robots are best at. When you put a robot into an unknown situation, however, odds are you're not going to have a design that's optimized for whatever that situation ends up being. This is where modular robots come in handy, since they can reconfigure themselves on-the-fly to adapt their hardware to different tasks. However, Modular Robotics Lab at the University of Pennsylvania has come up with a wild new way of dynamically constructing robots based on their CKBot modules: spray foam.

The process starts with a "foam synthesizer cart" that deploys several CKBot clusters, each consisting of a trio of jointed CKBot modules. The CKBot clusters can move around by themselves, sort of, and combined with some helpful nudging from the cart, they can be put into whatever position necessary to form the joints of a robot. The overall structure of the robot is created with insulation foam that the cart sprays to connect the CKBot clusters in such a way as to create a quadruped robot, a snake robot, or whatever else you want.

Having a robot that shoots foam is good for lots more than just building other robots. For example, Modlab has used it to pick up hazardous objects and to quickly deploy permanent doorstops. There's still some work to be done with foam control and autonomy, but Modlab is already thinking ahead.

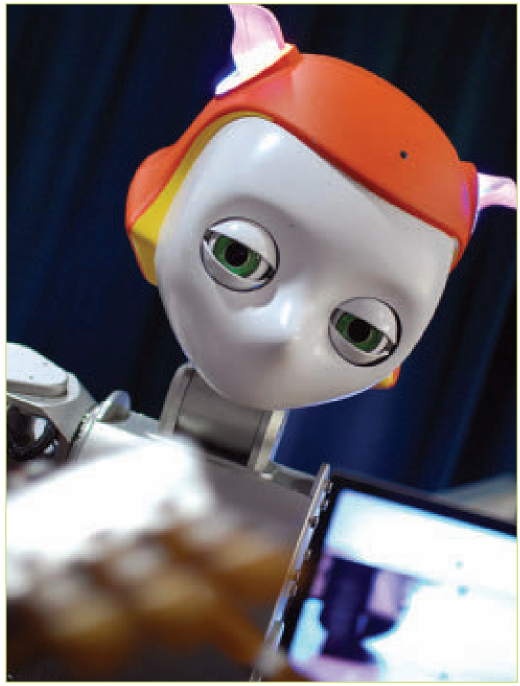
"By carrying a selection of collapsible molds and a foam generator, a robot could form end effectors on a task-by-task basis — for example, forming wheels for driving on land, impellers and oars for crossing water, and high aspect ratio wings for gliding across ravines. Molds could also be made of disposable material (e.g., paper) that forms part of the final structure. Even less carried overhead is possible by creating ad-hoc molds: making a groove in the ground or placing found objects next to each other."

With this kind of capability, you could send a bunch of modules and foam to Mars, and then create whatever kind of robots you need once you get there. With foam that dissolves or degrades, you could even recycle your old robots into new robots if the scope of the mission changes. Modular robots were a brilliant idea to begin with, but this foam stuff definitely has the potential to make them even more versatile.

THE EYES HAVE IT

Meka Robotics is based in San Francisco, CA, and they're probably best known for their underactuated, compliant hand (and the arm that goes with it), and now more recently for their humanoid head. The S2 head is notable because it manages to maintain a high degree of expressiveness (those eyes are amazing) while entirely avoiding the Uncanny Valley effect, thanks to its vaguely cartoonish look.

Meka is offering an entirely new system consisting of an arm, gripper, sensor head, and mobile base for $200,000. It's no coincidence that the one-armed PR2 SE costs the exact same amount; the NSF's National Robotics Initiative provides research grants, including up to $200k for research platforms. Yep, the government is basically giving these things away for free. All you have to do is convince them that you deserve one, and then pick your flavor.



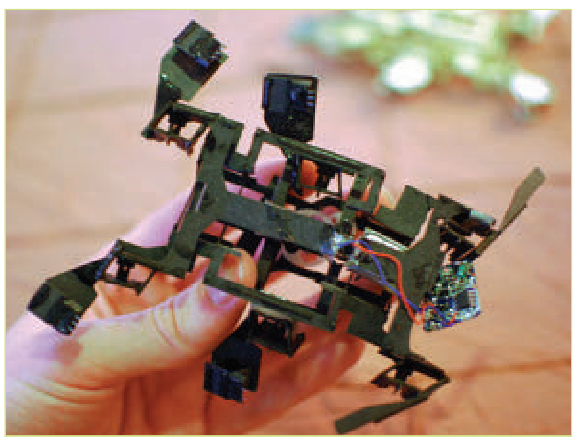
CLOTH CLIMBER

UC Berkeley has a long history of developing innovative legged robots. There was ROACH, BOLT, and there was DASH. DASH — a cockroach-inspired design — was a very simple, very fast hexapedal robot that could scuttle along the ground at 15 body lengths per second.

Now, meet the latest addition to this family of robot bugs: CLASH. CLASH is a vertically-enabled successor to DASH, and it's designed to zip up vertical or near-vertical cloth surfaces with the aid of tiny little spiny toes. It's sort of like what you'd get if you put DASH and SpinyBot together in a dark room along with a 3D printer and some Barry Manilow (or whatever it is robots are listening to these days).

For a vertical climbing robot, CLASH is surprisingly quick. It may actually be one of the quickest climbing robots in existence — it's able to move upwards at 24 centimeters per second, which is really a lot faster than it sounds.

Part of the reason that CLASH can scramble around so fast is that it's small and lightweight with a simple, but clever, design. CLASH is 10 centimeters long and weighs only 15 grams. The back-and-forth climbing motion of four legs (the back two are passive) is entirely driven by one single motor that gives CLASH a gait frequency of a brisk 34 strides per second.



The actual gripping and climbing technique is integrated into the beautiful series of linkages that connect CLASH's legs to its motor and to each other, making the mechanism completely passive all the way from initial grip to retraction. The battery and electronics are all onboard, and are located in the tail to help keep the robot balanced.

Next up is to endow CLASH with the ability to turn (which will likely involve the addition of a second actuator somewhere) and a modification of the rear legs to allow the robot to scamper along horizontal surfaces too. While CLASH is currently restricted to climbing things like fabric and carpet that it can sink its claws into, other methods of passive adhesion (like some of that gecko tape) might give CLASH a little extra versatility.

HERE'S THE POOP SCOOP

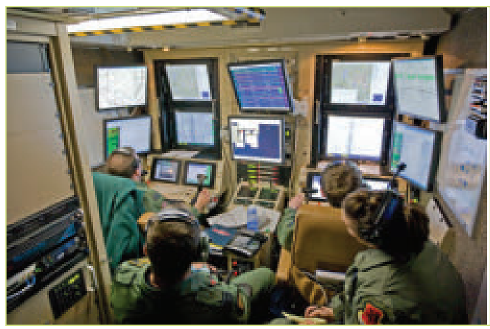
This could be it, folks. The one killer application that the entire robotics world has been waiting for. It's bold, it's daring, it's potentially transformative, and you know you want it: it's POOP.

Ben Cohen and his colleagues from the GRASP Lab at the University of Pennsylvania devoted literally an entire weekend to programming their PR2 robot, Graspy, to handle POOPs. POOPs (Potentially Offensive Objects for Pickup) are managed by the robot using a customized POOP SCOOP (Perception Of Offensive Products and Sensorized Control Of Object Pickup) routine. While POOP can be just about anything that you'd rather not have to pick up yourself, in this particular case, the POOP does happen to be poop, since arguably, poop is the worst kind of POOP.

Graspy begins its task by declaring in a vaguely disappointed robotic monotone, "time for me to scoop some poop." You get the sense that this $400,000 robot is asking itself whether or not this kind of work is really what it signed up for. Using its color camera, the robot first identifies poop based on its color, navigates to said poop, and then using a special human tool, performs the scoop. Haptics are employed to ensure that each poop scoop is a success, and if not, the robot will give it another try. Failure doesn't happen often, though. Graspy is able to successfully scoop poop about 95% of the time in over 100 trials, at a rate of over one poop per minute.

There's still some work to be done in order to get PR2 scooping poop like a pro (or an obedient human). For example, it's currently only able to handle high fiber poop, although that may be solvable with a different tool. If you think you have a clever way of making PR2 a better poop scooper, you can download the POOP SCOOP ROS stack and contribute to the betterment of humanity through robotics.

FLEET FLUISH

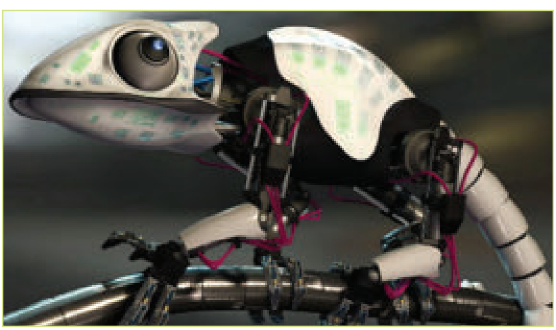


A virus has hit the military's UAV fleet. Details are sketchy as to its purpose, but it seems the key logger payload locks in keystrokes that drone operators perform. They have yet to be able to remove the virus and are not sure if its arrival was intentional or accidental. While it has not caused any obvious damage as yet, this certainly proves that nothing is totally safe in the Interweb world.

Note that security expert Miles Fidelman believes that it is possible that a rootkit that keys logs may have come from a DOD vendor.

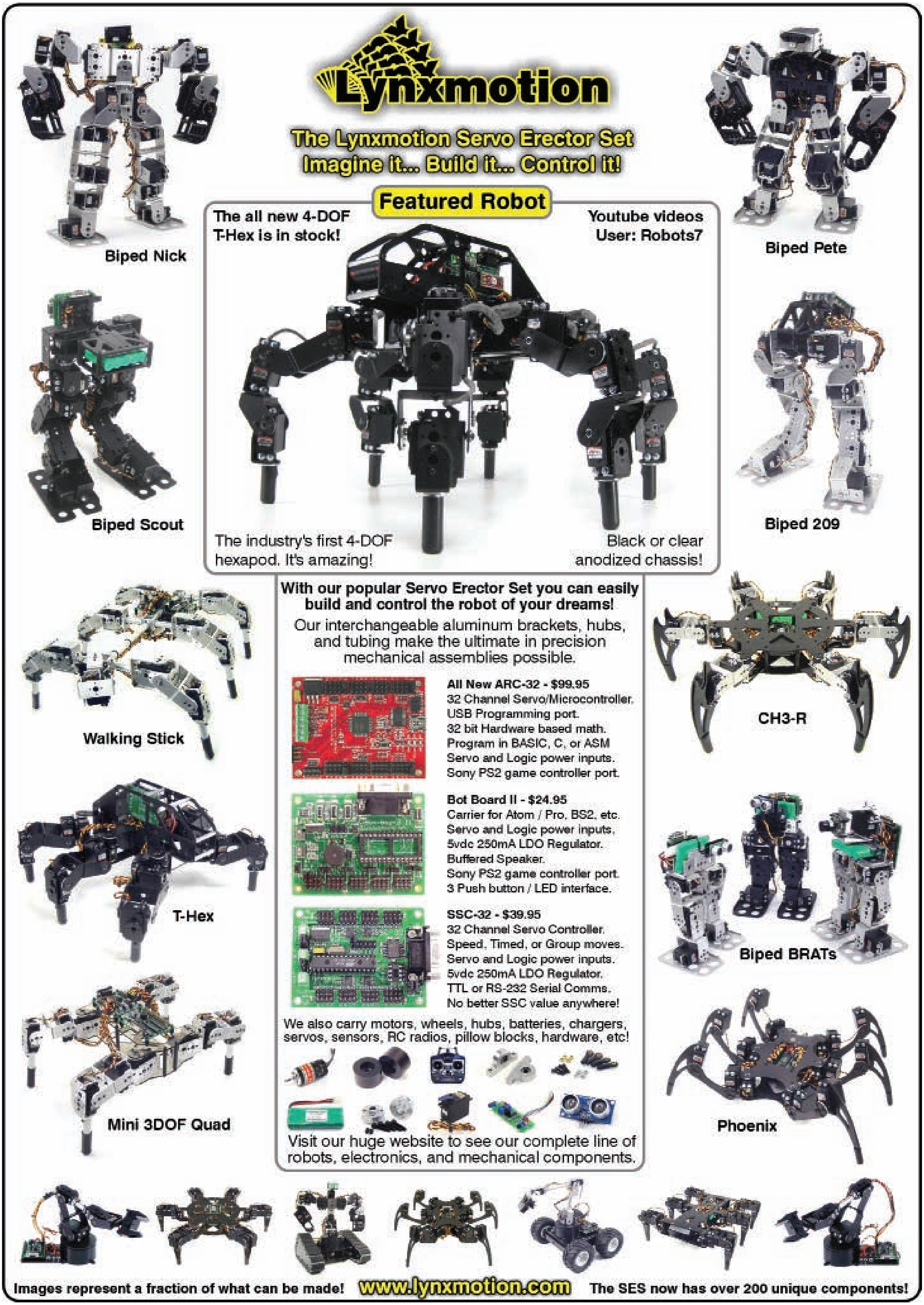
TONGUE IN CHEEK

Tomofumi Hatakeyama and Hiromi Mochiyama have not yet created a robotic chameleon like the one in the photo, but they have started in on one of the most important parts: the tongue. Chameleons can shoot their tongues out to capture prey in just three one-hundredths of a second, and then reel their tongues back in and chow down. Seems like a handy thing for a robot to be able to do, right?



This system is deceptively simple, relying on an air cannon of sorts to fire a magnetic projectile attached to a thin elastic cord. Over 90 percent of the time, the robotic tongue can snap up magnets dropped 0.7 meters away, taking barely a tenth of a second to traverse the distance and making the entire round trip in another tenth. It's wicked quick, and can nail almost exactly the same spot in mid-air every time.

Obviously, there are a few reasons why this particular version is probably not going to replace a real chameleon any time soon. For one, it only works on magnetic stuff and in order to make the catch, it needs a break-beam sensor to tell it when to fire. The researchers want this thing to ultimately shoot out to 10 meters (!), and they're planning to mount it on some kind of mobile robot platform that will scuttle around and catch cockroaches and other bugs.


Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
Featured Robot
The all new 4-DOF T-Hex is in stock!
Youtube videos User: Robots7
The industry's first 4-DOF hexapod. It's amazing!
Black or clear anodized chassis!
Biped Nick
Biped Pete
Biped Scout
Biped 209
With our popular Servo Erector Set you can easily build and control the robot of your dreams!
Our interchangeable aluminum brackets, hubs, and tubing make the ultimate in precision mechanical assemblies possible.
All New ARC-32 - $99.95
32 Channel Servo/Microcontroller.
USB Programming port.
32 bit Hardware based math.
Program in BASIC, C, or ASM
Servo and Logic power inputs.
Sony PS2 game controller port.
Bot Board II - $24.95
Carrier for Atom / Pro, BS2, etc.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
Buffered Speaker.
Sony PS2 game controller port.
3 Push button / LED interface.
SSC-32 - $39.95
32 Channel Servo Controller.
Speed, Timed, or Group moves.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
TTL or RS-232 Serial Comms.
No better SSC value anywhere!
We also carry motors, wheels, hubs, batteries, chargers, servos, sensors, RC radios, pillow blocks, hardware, etc!
Visit our huge website to see our complete line of robots, electronics, and mechanical components.
Walking Stick
CH3-R
T-Hex
Biped BRATs
Mini 3DOF Quad
Phoenix
Images represent a fraction of what can be made!
www.lynxmotion.com
The SES now has over 200 unique components!

COMBAT ZONE

Discuss this section in the SERVO Magazine forums at http://forum.servomagazine.com

Featured This Month:

www.servomagazine.com/index.php?/magazine/article/december2011_CombatZone

PARTS IS PARTS

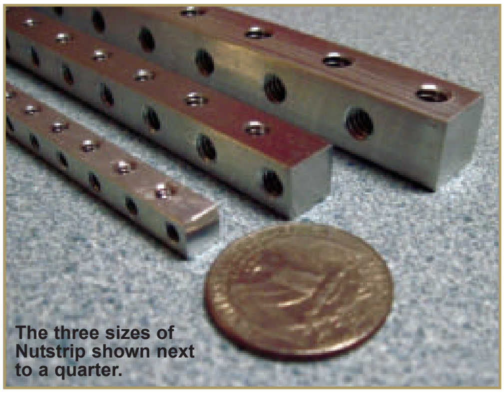
Kitbots Nutstrip

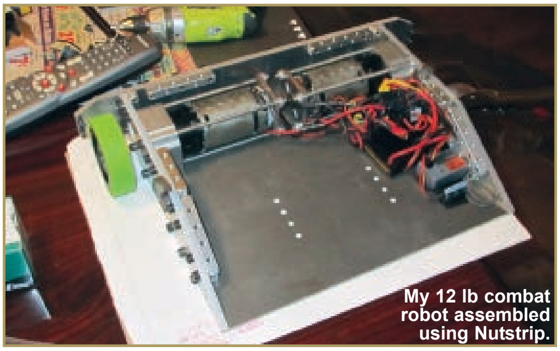
● by Mike Jeffries

Nutstrip is a product Kitbots uses in several of their kits to allow easy assembly and reduce machining costs on the chassis components. Nutstrip is a simple and effective design. The product is made from square stock and has a series of evenly spaced tapped holes accessible from all sides. Kitbots offers three sizes of Nutstrip: Mini (1/4" stock, 6-32 threads), medium (3/8" stock, 10-24 threads), and large (1/2" stock, 1/4-20 threads).

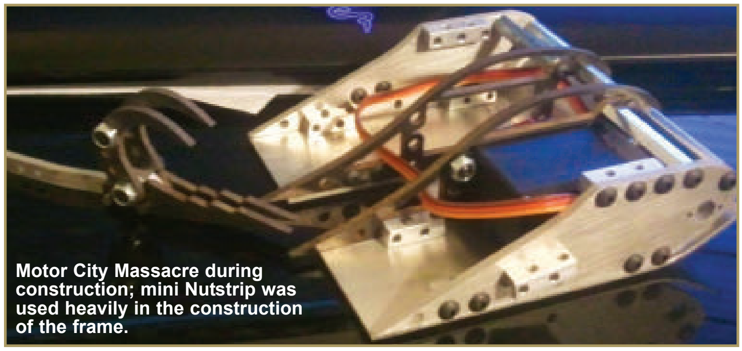
The primary benefits of using Nutstrip are that you only need to use through holes on your parts when you want to assemble them at right angles, and the threaded portion of the assembly can easily be replaced. This reduces the time and precision needed for most applications since threading takes additional time, and loose tolerances can be used with the through holes to allow for easy part fitting. In addition, if the threads are damaged the parts are much easier to replace.

I am currently using the mini and medium Nutstrip, and plan to use it as the primary means of assembling a 30 lb Sportsman class robot. In testing so far, the Nutstrip has held up to both the abuse of strong impacts and repeated fastener removal without any obvious signs of wear or damage. In my 1 lb robot, Motor City Massacre, the


The three sizes of Nutstrip shown next to a quarter.



My 12 lb combat robot assembled using Nutstrip.



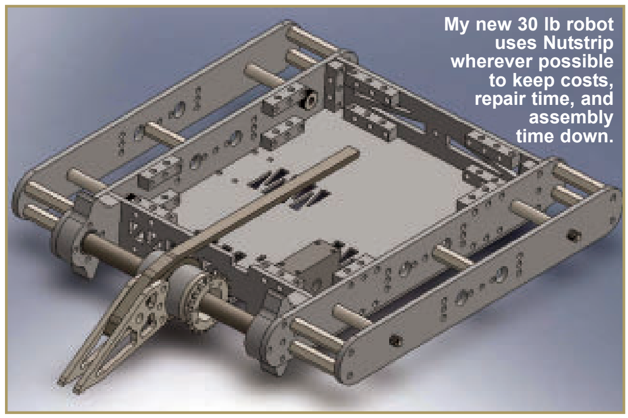
Motor City Massacre during construction; mini Nutstrip was used heavily in the construction of the frame.

Nutstrip is used to hold the chassis together along with threaded hex standoffs and to hold the thin steel cover over all of the electronics. At Dragon*Con Robot Battles, the steel cover was hit directly, taking a large chunk of material out of the armor and severely bending the side of the frame. During a post tournament repair, I determined that none of the Nutstrip on that side of the chassis sustained any damage and the only repairs needed were to the bent chassis member.

In addition to that, I've run my 12 lb combat bot through a robot hockey competition and the 12 lb class at Dragon*Con with no discernable wear or damage to any of the Nutstrip.

The positive results in testing have given me the confidence to begin work on that Sportsman class robot using the large Nutstrip as the main means of assembling the chassis. With the durability shown by the mini and medium Nutstrip, I am confident that the large version will be capable of handling the extra energy present in the 30 lb class.

By using Nutstrip in this build, I'm saving a great deal of time both in initial fabrication and repairs, as well as a great deal of money due to ease of fabrication and the reduced need for spare parts because both sides of the chassis can be assembled with identical hardware. SV



My new 30 lb robot uses Nutstrip wherever possible to keep costs, repair time, and assembly time down.

MANUFACTURING:

Designing for Waterjet

● by Mike Jeffries

When designing a robot, often the focus is on the weapon system or how much power you can cram into the drive system. Taking the time early on to think about how you'll make the chassis can heavily influence the design and be used to save money or time during a build.

In addition to the traditional machining processes of milling, turning, and drilling, there are processes that are considered non-traditional. These processes include waterjet, laser, and electrical discharge machining (EDM).

Each process has positives and negatives that must be considered, but when used correctly they can open up a wider range of design potential. While laser and EDM are both useful machining methods, the wide range of materials that can be cut using waterjet machining makes it a great option for experimenting with non-traditional processes.

Waterjet machining uses a very thin, high pressure jet of water, often with an abrasive material added to aid in cutting. This method of cutting means it is able to be used on a wide range of materials and thicknesses. Cutting with a fine jet of water also means that a small amount of material needs to be removed to cut the part to shape, reducing the machining time. Waterjet machining is a cold process so it eliminates the risk of heat-based part deformation.

The main drawback of waterjet machining is that you have to work exclusively with through holes for any portion of a part that will be cut with the waterjet. This drawback

Large batch of waterjet cut parts including steel, aluminum, and polycarbonate.

Moros assembled, showing waterjet cut panels with end-tapped holes for assembly.

can be worked around without too much trouble and several of the techniques of doing so will be discussed in this article. Using one or several of these techniques will allow you to design platforms that take advantage of the speed, accuracy, and relatively low cost of waterjet components.

Another key detail to remember is that waterjets cannot cut a perfectly square inside corner. The jet of water has a diameter, and will leave a small radius on the part. This can either be removed with a file or by having extra material cut out in the area to allow a square edge to move into the part without issue.

Techniques

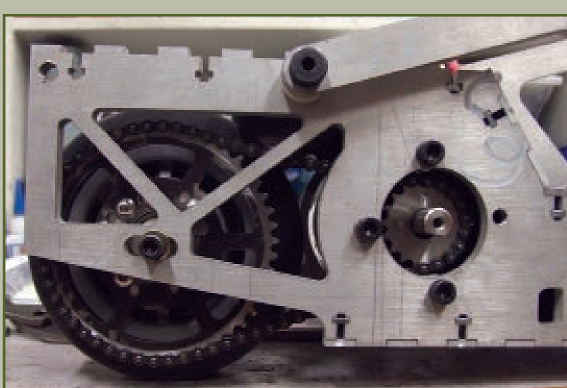
Welded Components

The first and lightest technique is to take the flat cut sheets and weld them together to form a three-dimensional frame. For this technique, you will want the parts to come together much like a three-dimensional puzzle. If the parts interlock well and the chassis can hold its shape without any welding being performed, this will make it more likely to retain the desired shape after welding has been completed.

For best results, you need to keep the chassis from getting hot enough to warp under the heat. While this is a very light chassis, the difficulty in keeping parts from warping during welding makes it a fairly difficult method of construction. Part replacement when using this method is cumbersome, as disassembly of the chassis is difficult.

Post Machining

The second technique is to post machine the parts that need additional features. After you have your pile of cut sheet, you can then do what is often done with other manufacturing methods and perform post waterjet machining to add tapped holes and other attachment features as needed. If you have an accurate way of marking tapped hole locations on the parts, this can be used to create a fairly light chassis. I used this method when building my 30 lb bar spinner, Moros.



Custom electric scooter using T slot and nut assembly. Built by Charles Guan.

The massive weapon system ate up a huge portion of the overall system weight. Using this method allowed me to create a chassis that weighed under 5 lbs that has proven to be capable of handling a massive 10 lb steel bar spinning at somewhere near 2,000 RPM. Part replacement is possible, however, if the part that needs to be replaced has not been through the necessary post machining steps, it could take a great deal of time.

T Slot and Nut Assembly

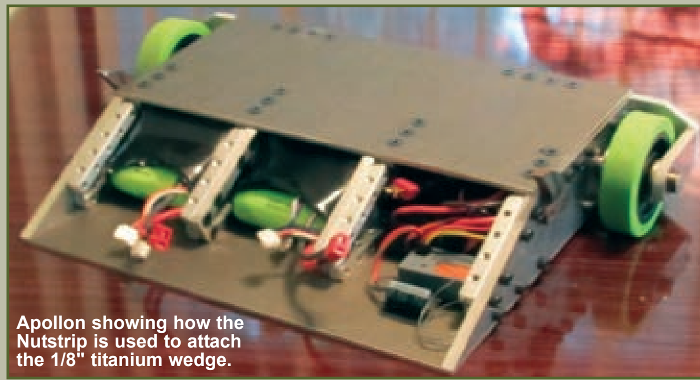
The third technique involves cutting the sheets with many T shaped slots to allow nuts to be slid into the chassis. This method of construction needs a great deal of forethought during the design stage, because the only resistance to side loads is the friction due to the load applied to the nut.

Designing chassis components that interlock to a degree will help resist side to side motion and can be used to sufficiently restrain the attached panels. Part replacement is very easy as the system can be quickly disassembled and reassembled without part modification.

Nutstrip

The fourth technique uses a product sold by **Kitbots.com** that they call Nutstrip (refer to the PARTS IS PARTS article this month). Nutstrip is a piece of square aluminum stock with a series of evenly spaced perpendicular tapped holes. This allows you to quickly and easily attach panels and form the structure of the

Motor City Massacre uses Nutstrip and threaded standoffs to hold the waterjet cut chassis together.



Apollon showing how the Nutstrip is used to attach the 1/8" titanium wedge.

chassis. I have used Nutstrip successfully in two robots: the Antweight "Motor City Massacre" and the Hobbyweight "Apollyon."

This use of a universal connector greatly simplifies design, construction, and maintenance. From a design standpoint, you only need to add the appropriate hole pattern to the designed parts to allow them to be connected to other portions of your chassis. With all the holes pre-tapped and the armor panels exclusively using clearance holes, there is little wasted time dealing with alignment and extra fixtures. If a part is damaged, spares can be made relatively cheaply since they need no post processing. If the Nutstrip is damaged, they are easily replaced. **SV**

Pretzel Robotics— Rookie Powerhouse

● by Chris Olin

September 2010. An unknown father-son robot team arrives unannounced and unexpected at their first competition (HORD 2010) with their new Antweight robot, Low Blow. Later that day, this mystery team drove home having won fourth place out of a field of nine, and were named "Rookie of the Year."

The following February, Low Blow's more evil twin, Vile Ant, took fourth place out of a field of 29 at NERC's Robot Conflict at Motorama 2011 (Low Blow finished 2—2 with the second loss being a forfeit to Vile Ant).

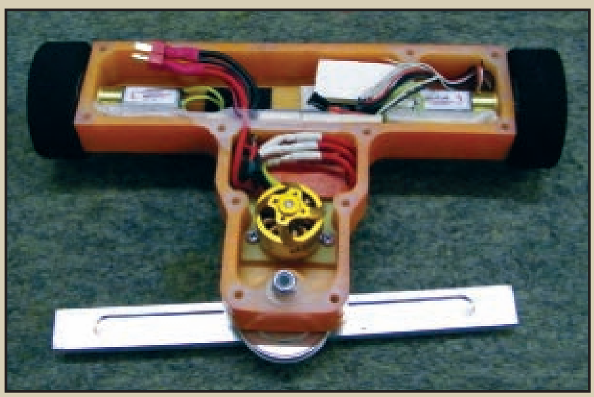
This past September, they returned to HORD and out of a field of nine ants, Vile Ant finished seventh while Low Blow finished second, losing the final match by a narrow judges' decision.

Who are these guys, where did they come from, and how the heck did a rookie team learn to build such effective robots?

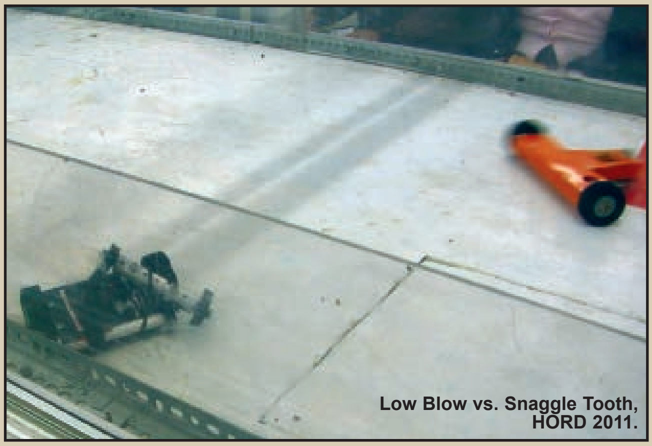
Glenn Purvin, the elder half of the team, is a mechanical engineer from Harrison Township, MI, with a strong background in the fastener industry. He describes his son Warren as a bright 13 year old. The family became interested in robotic combat while watching Battlebots on TV (2000-2002). In 2005, they attended Rochester Robot Rampage and met many of the big names in robot combat. That is when Warren first wanted to build his own robot, but it was decided he was too young and would have to settle for Pinewood derby cars and the like.

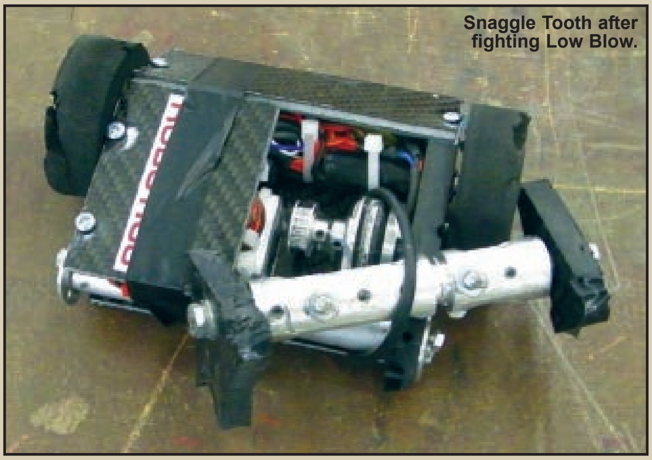
Plans for their own Antweight robot began in early 2010. Glenn and Warren combed through back issues of *SERVO Magazine*, making note of the success of Thomas Kenney's fast and low wedge Gilbert, and used the article "The Intro Ant" (*SERVO*, June '08) as a guide for selected components. They also read "RioBot Tutorial" which suggested a horizontal spinning blade would be

Low Blow's insides. Weapon: Turnigy brushless weapon motor and controller with FingerTech timing belt and pulleys driving an aluminum blade. Drive: FingerTech Spark gear motors and hubs driving Lite Flight Wheels. Power: Rhino Lipo batteries and Spektrum BR6000 receiver.




Low Blow vs. The Froogin, HORD Fall 2011.


Low Blow vs. Snaggle Tooth, HORD 2011.


Snaggle Tooth after fighting Low Blow.


Low Blow vs. Meerkat Mreow, HORD 2011 final.

effective against a fast and low wedge.

With their research complete and a general plan in mind, it came time to do the CAD (Cardboard Aided Design) work. The basic layout was drawn on graph paper, then transferred to a piece of 1/4" foam core display board. The components were then strapped down with zip ties. This allowed them to work out the wiring and confirm functionality of the drive system. It also provided a close estimate of the weight requirements for the body and weapon.

UHMW was selected for the bulk of the body, based on its low density and their lack of proper metal working machinery. They quickly learned why this material is highly desirable as armor — namely, that it is resistant to sawing, grinding, etc.

After cutting the material with a scroll saw proved ineffective, they made patterns from the same 1/4" paneling and used a router to carve out the desired shape. Cutting the patterns took longer than cutting the final body, but it did allow them to quickly cut another body for Vile Ant later on. Garolite top and

Low Blow vs. Kyle's Cutter, HORD 2011.


Glen and Warren Purvin, HORD Fall 2011.

bottom plates completed the chassis assembly.

Once the chassis was finished, the components were mounted and fine-tuned. The result was a highly effective under-cutter bar spinner.

Expect to see Low Blow and Vile Ant continue climbing the Antweight ranking. SV

DRILL BABY, DRILL. Or, the Good, the Bad, and the Ugly

● by Pete Smith

For about a decade or more, it has been possible to get cheap cordless drills from Harbor Freight (**www.harborfreight.com**). These have steadily improved in power over that time, with the standard voltage climbing from 9.6V through 12V and 14.4V to 18V — all with essentially the same size gearbox and motor RPM.

The motors and gearboxes from these drills have powered many combat bots over the years in both the 12 lb and 30 lb weight classes.

However, this year Harbor Freight has changed its supplier and the old range is no longer available. My whole fleet of Bot Hockey bots and my 12 lb combat bot "Surgical Strike" used these motors, so I started a quest for a suitable replacement.

My requirements were:

- 14.4V–24V nominal voltage.
- 550–1,000 RPM at nominal voltage.
- Single speed (the two speed gearboxes tend to be larger and heavier).
- Standard gearbox (i.e., like the Harbor Freight models) form factor.
- 500 sized motor.
- Standard 3/8-24 NF output shaft.
- "Double D" nose on the gearbox.
- Less than $25 each.

Much "Googling" later, I found three possible candidates.

The first was a new drill by Harbor Freight (**Figure 1**) — their Drillmaster model 68239. The second was the 18V Power Smith model from Northern Tool (**Figure 2**). The last was one advertised as being by Boston Industrial (**Figure 3**) on Amazon.

I dismantled all three as per my article in the November '06 issue of *SERVO*, and they all came apart without major problems. I will cover the details of each drill in turn.

The Good

First, the Harbor Freight 68239. The case of this drill has one screw hidden behind a label (**Figure 4**), but you can easily pierce the label to get at it.

The pinion gear on the motor (**Figure 5**) is a press fit and smaller than that of the older Harbor


FIGURE 1


FIGURE 2


FIGURE 3


FIGURE 4

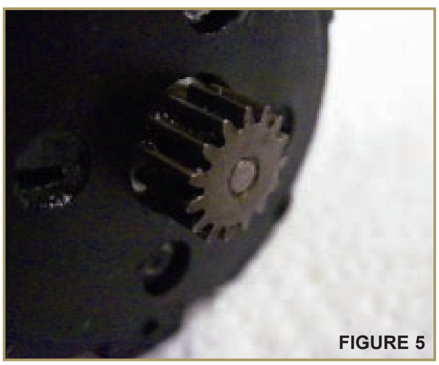


FIGURE 5

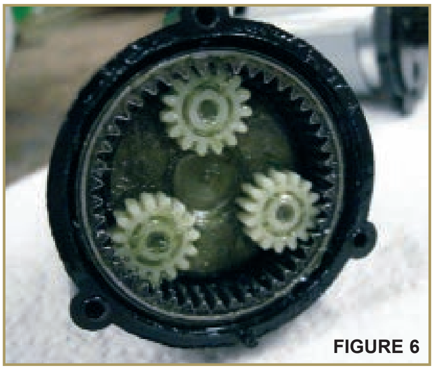


FIGURE 6

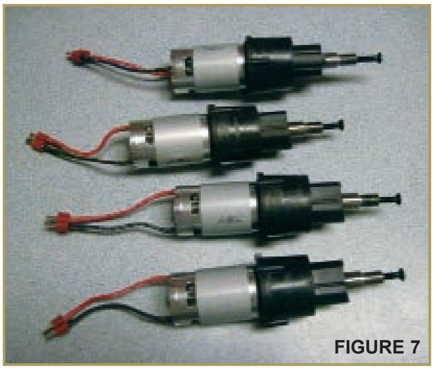


FIGURE 7

FIGURE 8

FIGURE 9

FIGURE 10

Freight models so it could not be used as a spare for them. The first stage gears are plastic (**Figure 6**) but the second stage are metal. The different motor and gears give a higher output speed of 900 RPM.

The casing is very similar to the old drills, but the Double D nose is just slightly larger so will require gearbox mountings to be modified. The output shaft is 0.3" longer (**Figure 7**).

To sum up, this drill should be good for combat use in a 12lber with 3" wheels.

The plastic first stage is a drawback, but most impacts will be on the second stage metal gears. The longer shaft, increased RPM, and slightly larger case means they are not a direct replacement but could be fitted to most existing bots with a little modification of the mounts and the wheel hubs.

The Bad

The "Boston Industrial" drill purchased through Amazon looked at first glance to be nearly identical to the old Harbor Freight models, and I had hoped that they would be simply the same drill repackaged for a different vendor (none of the drills here are actually made by the companies selling them). However, this quickly seemed to not be the case since the batteries were not interchangeable.

Closer examination of the drill revealed all was not as it should be. The drill packaging and casing is labeled 18V (**Figure 8**) but curiously the charger was only rated at 9V (**Figure 9**).

The final answer came when opening up the battery pack to find only six cells, i.e., 7.2V (**Figure 10**) and the motor on the drill had a 7.2V sticker on it (**Figure 11**). Yes, it was a 7.2V drill, mislabelled and being sold as a 18V model! They say you only get what you pay for but it seems sometimes you don't even get that.

Amazon and the vendor have been contacted, and the drill had been removed from sale at the time of this writing.

The Ugly

The Power Smith isn't really ugly. It's as good a drill as the Harbor Freight model and it has the same 550 RPM. It also has plastic — but chunky — first stage gears (**Figure 12**) and a press fit gear. What makes it ugly for our purposes is that the gearbox case is quite different, with a rounded triangular nose (**Figure 13**). It would not fit any existing bot designed to fit the old designs. I've seen this triangular nose before but its not a common design, so future replacements may be even harder to locate.

Conclusion

The new Harbor Freight 68239 drill is usable as a drive train for 12 lb bots. Its first stage plastic gears are not as good as the metal ones we are used to, but these are not particularly likely to fail. The higher RPM should give

FIGURE 11

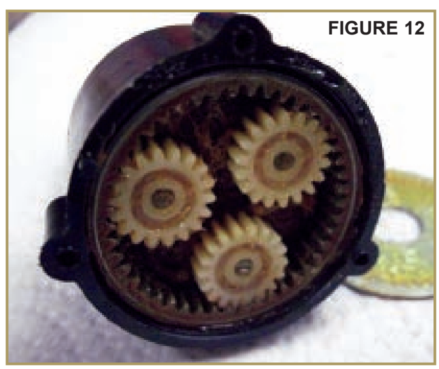


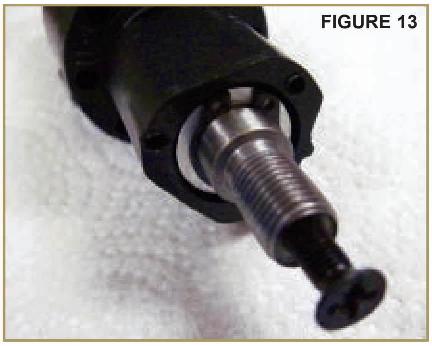
FIGURE 12

useful extra speed in bots with 3" or smaller wheels, but it comes with the cost of lower torque and a higher risk of stalling the motor. The slightly larger case is easily fixed with a file. The higher gearing might make them impracticable in a 30 lb bot.

The other two drills are a useful lesson in never assuming that a drill will be a suitable replacement until you have bought one and checked it out. The Power Smith could have been used in a new bot or even put back together, and it would have served as a reasonable light-duty drill and driver. The 7.2V fake serves as a good reminder of that wise advise, "Caveat Emptor."

The search for the perfect replacement cheap drill will apparently continue ... SV

FIGURE 13



EVENTS

Completed Events for September 2011

Seattle Bot Brawl 2011 was presented by Western Allied Robotics in Seattle, WA on September 11, 2011.

HORD 2011 was presented by the Ohio Robot Club in Brunswick, OH on September 24, 2011. SV

EVENT REPORT:

ORC Storms "The Gate"

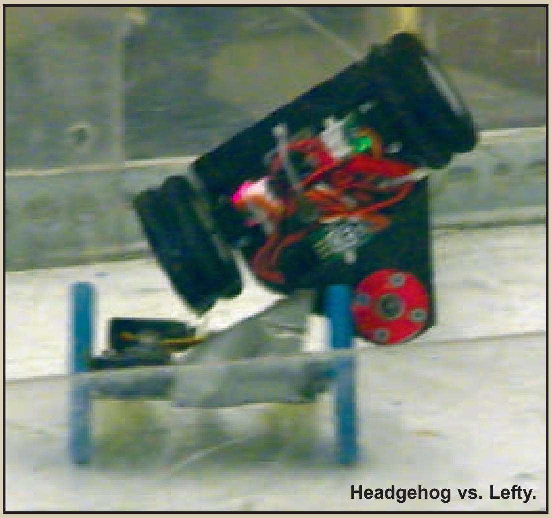
● by Chris Olin

2011 was a difficult year for the Ohio Robotics Club (ORC). First, their spring event at Cuyahoga Valley Career Center was canceled due to lack of support from the school and several personal schedule conflicts. Then, in early August, it was learned that the venue for their September event — Classic RC Raceways — had gone out of business. With no venue for an event scheduled to be held in little over a month, the ORC team scrambled to find a new venue.

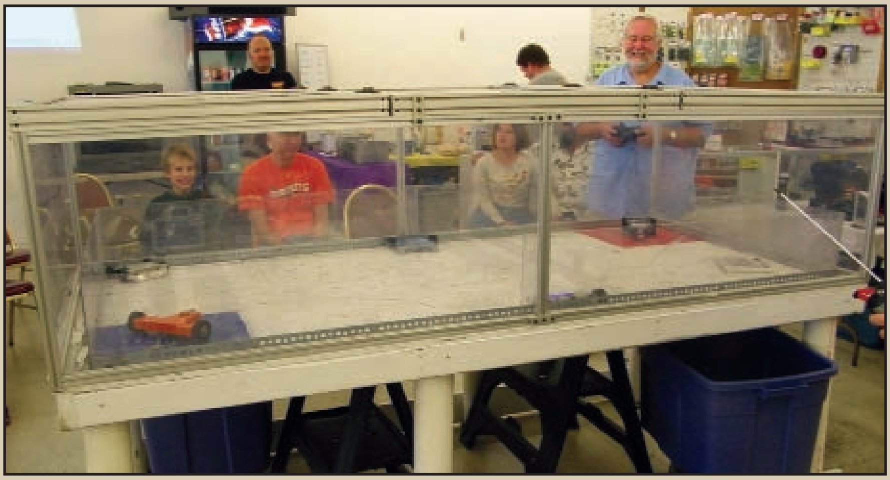
Through this adversity came a new opportunity. The Northern Ohio Radio Control Auto Racers (NORCAR) graciously offered ORC the use of their facility "The Gate" located in downtown Brunswick. The Gate features a large carpeted racing surface and spacious pits in a well-maintained storefront located in the Laurel Square Shopping Center. However, there was a catch. The Gate was not available for use on the originally scheduled date, (September 17th) so the event would have to be postponed one week. The issue was put to a vote and all registered teams agreed to the new place and time.

"House of Robotic Destruction 2011: Storming The Gate" was held on September 24th. Nine teams brought 20 robots to ORC's first (of hopefully many) event at this new location.

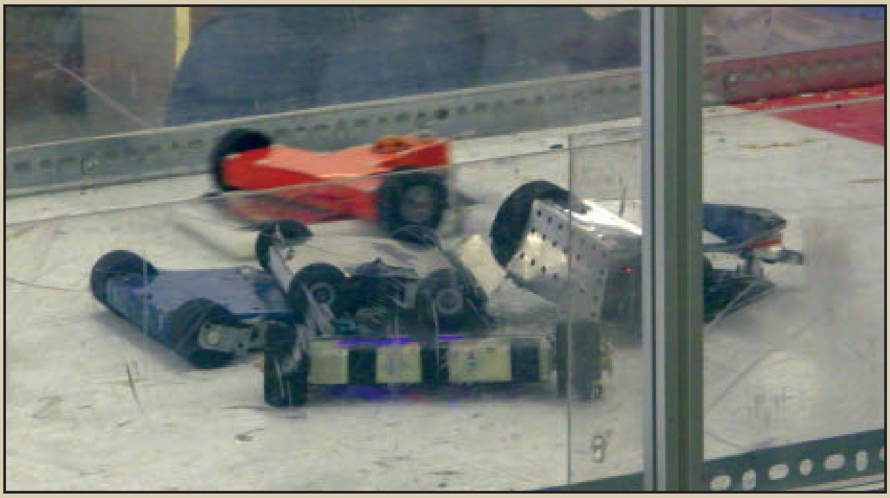
A field of five Fleaweights (150 grams) fought a round robin



Headgehog vs. Lefty.



Ant Rumble. Meerkat Mreow, The Froogin, Kyle's Cutter, Low Blow, Nyan Cat, and Bully.



tournament dominated by two horizontal spinners. David Gram's Headgehog and Don Jenkins' Irregular Pentagon each won three matches. First place was decided by a tie breaker match, won by Hedgehog by a narrow judges' decision. Rounding out the top three was Zach Witeof's pushy bot, Chairman Meow, with two wins.

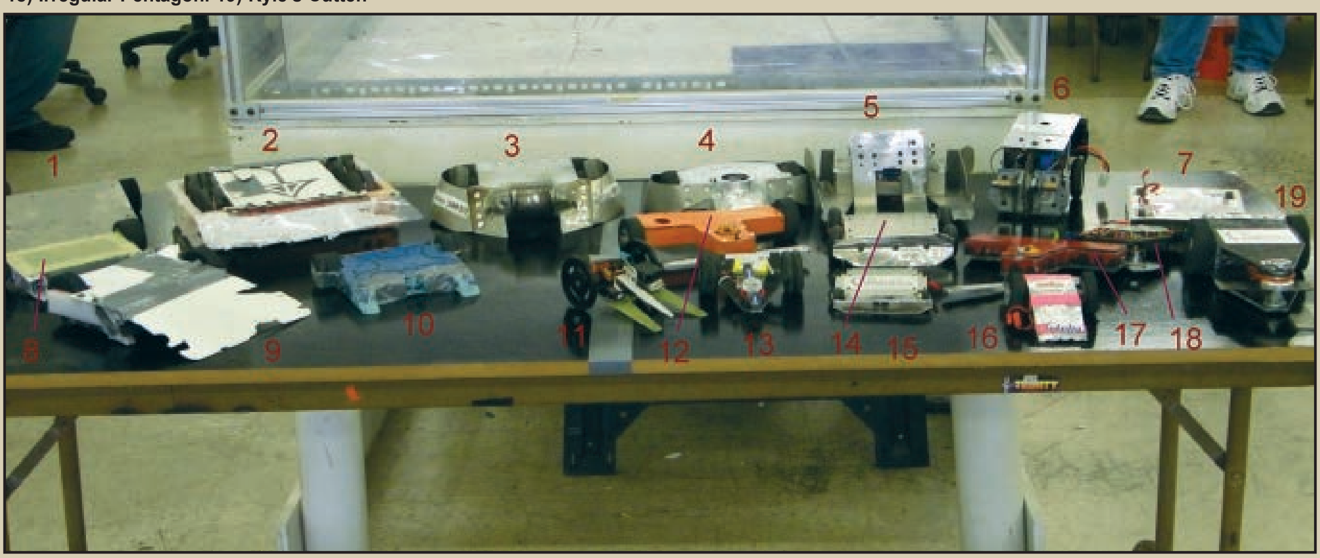
Meanwhile, nine Antweights battled through a grueling double elimination tournament. Zach Witeof's excellent driving of his four wheel pushy bot, Meerkat Mreow, took him all the way to the winner's bracket semi-final where he faced last year's "Rookie of the Year" Warren Purvin and his hard hitting undercut spinner, Low Blow.

In early rounds, Low Blow crimpled David Gram's drum spinner, Snaggle Tooth. Gram was unable to repair Snaggle Tooth and was forced to forfeit his next match. Low Blow knocked Meerkat down to the loser's bracket where he faced and defeated Alex Udanis' cheese wedge, "Trolling!" who had to settle for third place.

The final was a rematch of Meerkat and Low Blow. Low Blow suffered from a battery issue and was unable to press the attack, thus allowing Meerkat the victory and first place. Low Blow took second, improving over last year's fourth place.

Beetleweight action consisted of a six robot round robin including

WINNING ROBOTS: 1) Trolling. 2) Sweever. 3) One Fierce Lawn Boy. 4) One Fierce Round House. 5) Revenge of Dr. Super Brain. 6) Rippy. 7) Bully. 8) Nyan Cat. 9) Nix. 10) The Froogin. 11) Lefty. 12) Low Blow. 13) Headgehog. 14) Meerkat Mreow. 15) Chairman Meow. 16) Brutus. 17) Vile Ant. 18) Irregular Pentagon. 19) Kyle's Cutter.



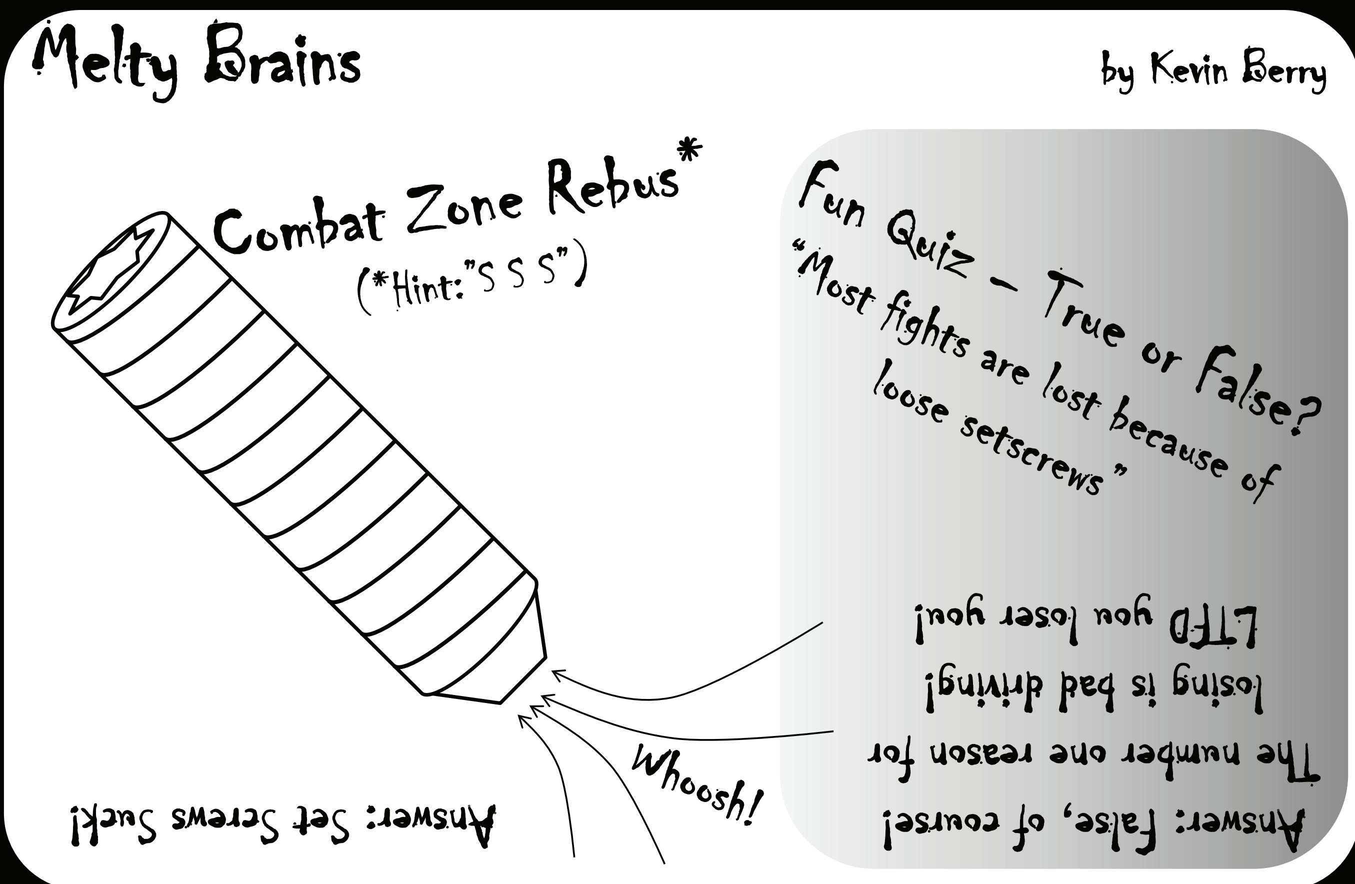
WINNING DRIVERS (left to right) – Back Row: Gene Burbeck, Chris Olin, Zach Witeof, David Gram, Warren Purvin, Don Jenkins, Alex Udanis. Front Row: Chad Savage and Caitlin Wilson.

Gene Burbeck's infamous two engines of destruction, One Fierce Round House and One Fierce Lawn Boy. Round House's horizontal disk cut through the ranks, while Lawn Boy's drum beat down all comers until only Chris Olin's redesigned and rebuilt servo powered lifter, Revenge of Dr. Super Brain, remained.

In round one, the Doctor faced Lawn Boy and used its lifter to high-center Lawn Boy on its rear plate; the immobilized Lawn Boy was counted out. Later in round four, the Doctor fought Round House; the two robots fought fiercely for about two minutes, then a particularly hard hit sent the Doctor recoiling and Round House tumbling into the pit. The Doctor limped to first place with five wins; Round House took second with four wins; and Lawn Boy took third with three wins (Lawn Boy forfeited to Round House).

Prizes and other considerations were provided by Dimension Engineering, FingerTech Robotics, and *SERVO Magazine*.

ORC would like to thank NORCAR, The RFL, their sponsors, volunteers, and competitors for helping put together a great event. For more information on future ORC events, see their website at **www.ohiorobotclub.com**. SV



Infinitely Modifiable.
MINDS-i
Extraordinarily Durable.
All-Terrain Robotics.
PRO-LINE
Experience the Future at mymindsi.com

The Robot MarketPlace
Have a developing robot enthusiast on your gift list?
CHALLENGE YOUR MIND
with educational robotics kits for all skill levels.
• Surveillance Robots
• Combat Robots
• Full Line of R/C Cars, Trucks Boats, Airplanes, & Helicopters
• Hobby R/C Accessories
Visit us online to view over 15,000 robot parts and hobby accessories!
(877) ROBOT 99
(877-762-6899)
Your Headquarters for R/C Hobby & Robot-Related Gifts!
www.RobotMarketPlace.com

EXPERIMENTING WITH UNCONVENTIONAL WHEELS

by John Blankenship and Samuel Mishal

www.servomagazine.com/index.php?/magazine/article/december2011_Blankenship

A robot equipped with omni or mecanum wheels has mobilities far different from robots with standard wheels. A LEGO NXT-based robot equipped with unconventional wheels can provide an easy-to-use, relatively inexpensive platform for experimenting with these capabilities. Find out if your next robot needs new wheels.

FIGURE 1. Two small omni wheels and a mecanum wheel.

Mobile robot bases typically utilize wheels, but there are numerous options and configurations. Typical bases might use four wheels or two wheels with one or two casters. The use of omni wheels or mecanum wheels (see **Figure 1**) can provide additional options.

Both omni and mecanum wheels can make it easy for a robot base to move in a variety of directions but in very different ways. A mecanum wheel consists of numerous rollers angled at 45º while the rollers on an omni wheel are mounted perpendicular to the wheel's normal rotation. Omni wheels are generally used in a

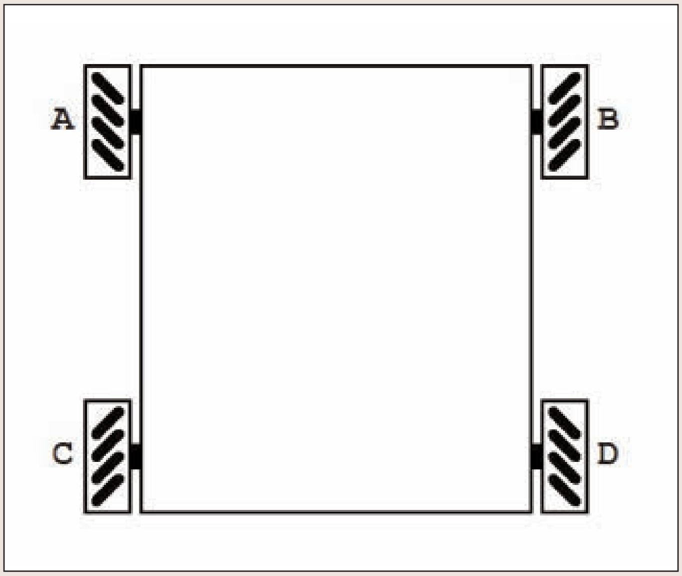


FIGURE 2. Mecanum wheels must be paired to perform properly.

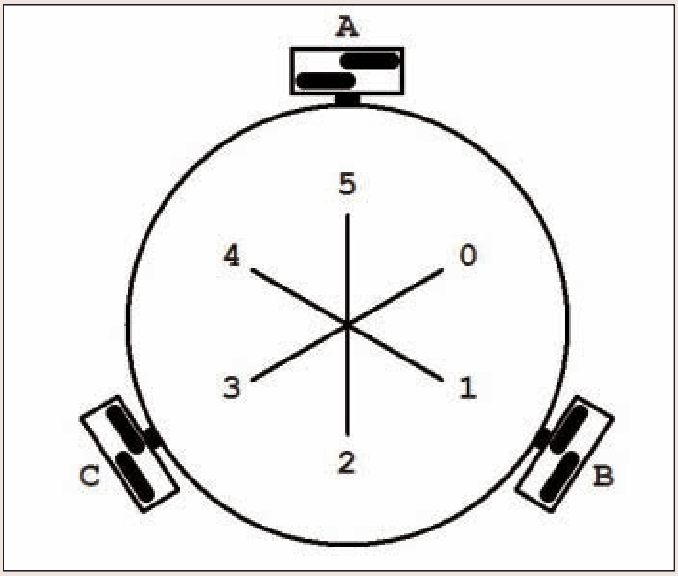


FIGURE 3. Omni wheels should be mounted at three equally spaced positions.

three-wheeled configuration while mecanum wheels are used in pairs to create a four wheel base. Both of these wheel designs can be relatively expensive because of the labor generally required to produce them.

When utilizing mecanum wheels, each wheel is a mirrored image of its apposing partner as shown in **Figure 2**. When all four wheels move in the same direction, the robot will move forward or backward. When wheels A and C move in the opposite direction of B and D, the robot will rotate around its center.

If the tops of wheels A and C move toward each other and the tops of wheels B and D move away from each other, the robot will slide to the left. Similarly, reversing these wheel motions will cause the robot to slide to the right. Movements at 45º angles can be obtained by turning only the two wheels at opposite corners in the same direction. All of these motions assume that the active motors are moving at the same speed. Using differing speeds will result in similar motions but at a variety of angles and orientations.

Figure 3 shows an omni wheel configuration. When all three wheels turn in the same direction, the robot will rotate around its center. In order to understand how to move the robot forward, we need to decide on where the front of the robot is. Assume that the front of the robot is direction 0. We can move the robot in that direction by turning wheel A forward and wheel B backward (making both motors turn in the same direction, referenced to the robot itself) at the same speed. Wheel C does not have to be activated since it acts as a caster for this motion. Reversing the directions of wheels A and B will move the robot in direction 3. If wheels A and B are not moving at the same speed, the robot will tend

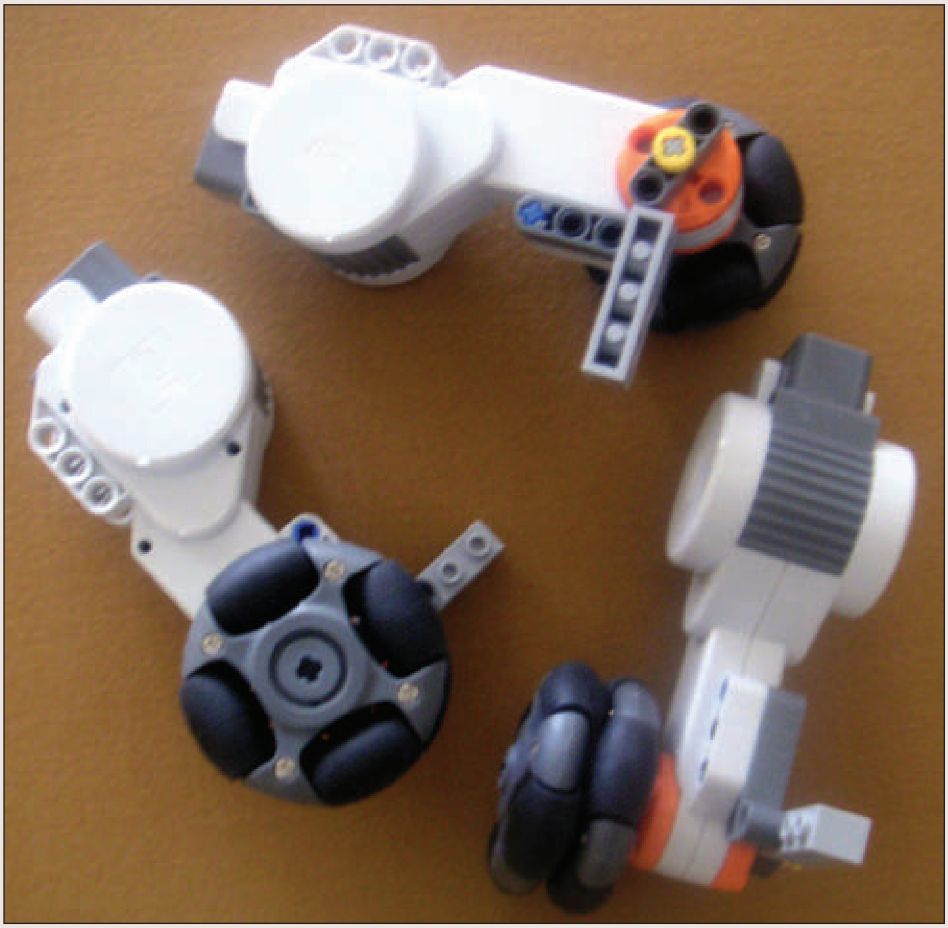


FIGURE 4. Omni wheels from Nexus Automation attach easily to LEGO motors.

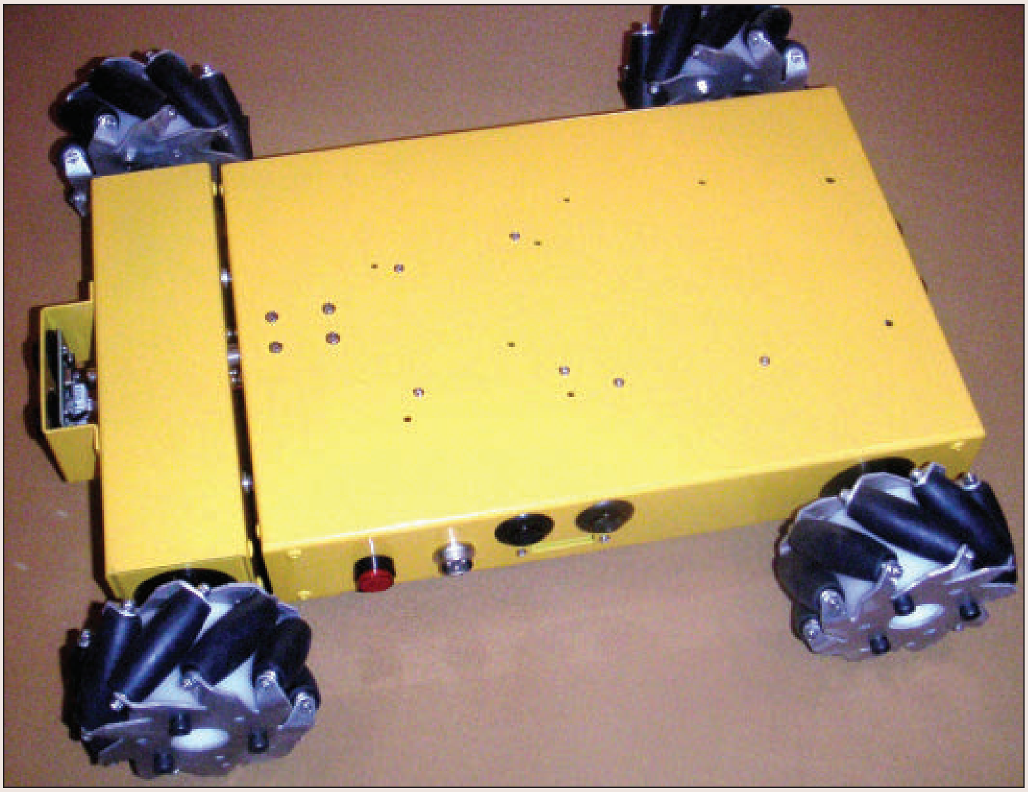


FIGURE 5. This Nexus robot utilizes mecanum wheels and supports sonar sensors on all four sides. Notice also that the front wheels have a suspension independent of the rear wheels, giving it better traction in some situations.

to drift either left or right, depending on which wheel is moving the fastest.

Using a similar philosophy, the robot can be moved in any of the six directions shown by simply choosing which two wheels to energize. These six directions are special because the chosen direction will remain the same, even if the speed changes. For example, if you double the speed of both motors the robot will just move faster in the designated direction. This linear relationship is unique to the six directions shown.

With experimentation, you can find a combination of motors and motor speeds to move the robot in almost any direction. If you double (or half, for example) the speed, the direction will also change slightly, often moving the robot in an arc rather than a straight line. This non-linearity generally means that a three-wheel omni robot is easier to control for some applications if you restrict its movement to the six linear directions.

You may feel like confining the robot to six directions is too restrictive, but remember, as long as you are willing to accept a single speed for any given direction, you can create the desired motion by experimenting with various motor speeds until you achieve your goal. This can be an interesting activity for many hobbyists, especially if they have never used multi-directional wheels. For such experimental purposes, it would be nice to have an easy and inexpensive way of building a usable robot base. In addition to the mechanical assembly, each motor must have the electronics to control both speed and direction. It would also be nice — especially for quick experimentation — to be able to write the control programs in a high-level language.

Since LEGO NXT motors have built-in speed control circuitry and the LEGO system can control three motors, it seemed like an excellent candidate for experimenting with omni wheels. Unfortunately, we found it difficult to find multi-directional wheels that would mate with the NXT motors. We discovered an emerging company in China called Nexus Automation (see **www.NexusRobot.com**) that offers omni wheels that attach directly to LEGO NXT motors as shown in **Figure 4**.

Nexus also offers mecanum wheels, as well as robot kits that utilize both types of wheels (see **Figure 5** for an example of one of their kits). Complete kits such as these

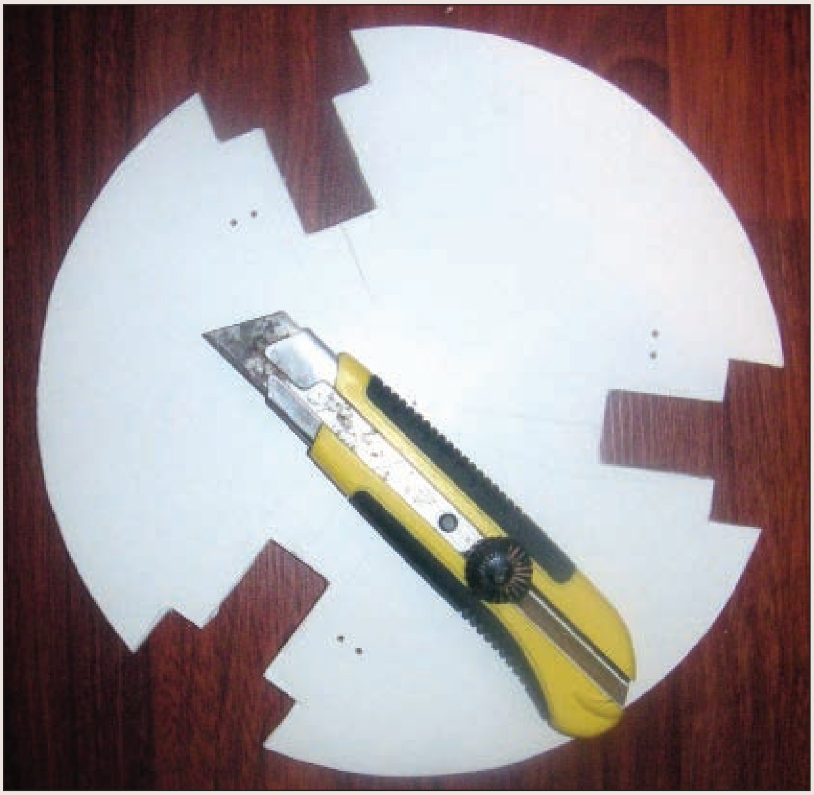


FIGURE 6. Foam board can be cut as needed with only a knife or razor blade.

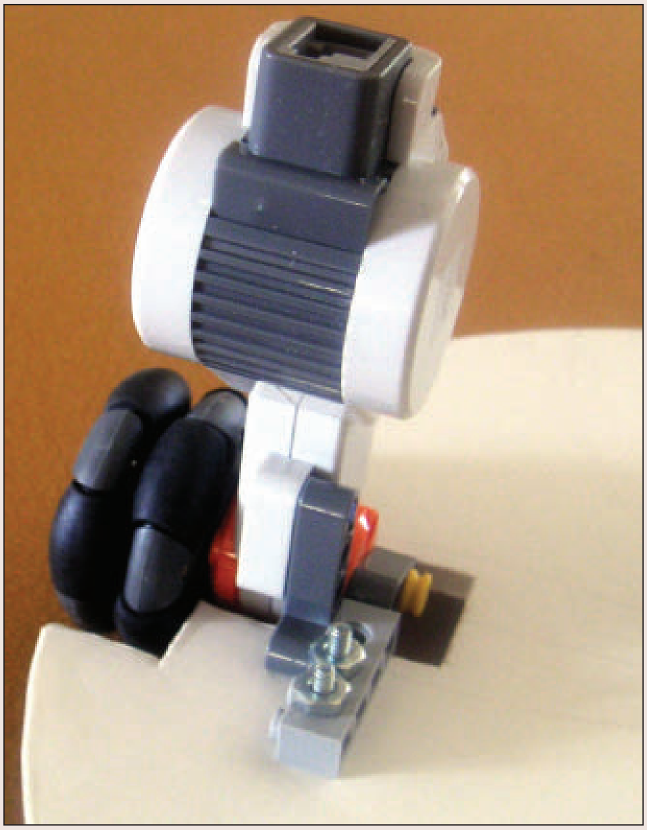


FIGURE 7. LEGO parts make it easy to use bolts to mount motors to the foam-board base.

```
sub LegoMotors(A,B,C)
    SerialOut 12,0,0x80,4,0,A,1,0,0,0x20,0,0,0,0
    SerialOut 12,0,0x80,4,1,B,1,0,0,0x20,0,0,0,0
    SerialOut 12,0,0x80,4,2,C,1,0,0,0x20,0,0,0,0
return
```

FIGURE 9. This RobotBASIC function provides speed and direction control for all three NXT motors.

can be expensive but worthwhile if you *know* you want a robot with one of these wheel configurations. This makes having an inexpensive experimental platform even more valuable because it can help you decide if multi-directional wheels are appropriate for your projects.

Nexus has just started looking for distributors here in the US so hopefully their products will be available soon without the high cost of international shipping. If you can't find a local source for their products, their marketing executive Anny Kong (Anny@NexusRobot.com) has agreed to provide special pricing for *SERVO* readers that *includes* shipping to the US. Three omni wheels are $50; four mecanum wheels are $260; and the robot kit shown in **Figure 5** is $1,450.

If you have a LEGO NXT system, you can easily mount Nexus' omni wheels (or perhaps similar wheels from other sources) to each of three motors. Rather than trying to create a base solely from NXT parts, we chose to use foam board as shown in **Figure 6**. Foam board can be purchased at nearly any craft store, is easily cut with a razor blade or hobby knife, and can be glued to quickly create relatively strong structures.

The NXT motors can be mounted to the foam board as shown in **Figure 7**. We chose to orient the motors vertically, but foam board is so easy to work with that you can let your imagination run wild. Vertical foam board beams were added next to each motor to provide lateral support for the motors and to act as legs for an upper level.

FIGURE 8. The finished robot has an ultrasonic distance sensor and three line sensors.

Our finished robot is shown in **Figure 8**. The second level supports the NXT brick (the computer) and — because it is glued to the vertical foam board supports — it adds significant rigidity to the assembly. Notice that we added an ultrasonic distance measuring sensor and three line sensors to make it easy to experiment with a variety of environmental situations.

You could use Mindstorms — LEGO's graphic oriented language — for your experimentation if you prefer, but we used RobotBASIC (a free language available from **www.RobotBASIC.com**) to interface with our robot. One of the nice things about the NXT is that communication with the motors and sensors can be handled using direct commands. This allows RobotBASIC programs to control the robot (and read sensor data) without downloading any programs to the robot itself.

LEGO's direct commands are simply strings of data that communicate directly with the NXT's

FIGURE 10. This program allows our omni-wheeled NXT to follow a wall.

```
// initialization
#include “OmniLegoLibrary.bas”
LegoPort=34
RangePort=4
call LegoInit(LegoPort)
call LegoRangeInit(RangePort)


// move forward till a wall is found at 5 inches away
repeat
    call LegoRangeSensor(RangePort,D)
    call LegoForward(SLOW)
until D<5

// turn robot slightly to the right to position it parallel to wall
Call LegoDegreesRight(30)
// and establish a new forward orientation
// with range sensor angled left
LegoDir = 1

// then follow the wall
while true
   call LegoRangeSensor(RangePort,D)
   if D<8
     gosub LegoSlideRight // moves right and forward
   elseif D>9
     gosub LegoSlideLeft  // moves left and forward
   else
     call LegoForward(SLOW)
   endif
wend
end
```

operating system over a Bluetooth link (refer to LEGO's documentation for information on establishing the wireless link). **Figure 9** shows a simple RobotBASIC function for controlling all three NXT motors. Larger parameters increase the motor speeds and negative numbers reverse the directions. You could create a similar routine with nearly any language capable of establishing a Bluetooth connection with the LEGO robot.

The function in **Figure 9** is only one of many functions we created as a library for experimenting with an NXT robot equipped with omni wheels. For example, the library provides functions to turn the robot and to move it forward. When using the library, the variable LegoDir specifies which of the six possible orientations (see **Figure 3**) should be considered the robot's forward direction. The default is 0 which points our robot's range sensor forward.

The program in **Figure 10** demonstrates the flexibility of using omni wheels. The program moves the robot forward until a wall is encountered at five inches. Once the wall is found, the robot turns 30 degrees to the right and chooses a new LegoDir of 1 — ideally making the robot somewhat parallel with the wall. The key to this direction change is that the ultrasonic range sensor now angles leftward (instead of forward, as it does for direction 0), allowing the sensor to be used to determine the distance to the wall or obstacle to the left of the robot.

Additional library routines are used to follow the contour of the wall by making the robot slide side to side when the robot is too close or too far from the wall. Each of the library routines perform their actions based on the current value of LegoDir. This simply means that routines like LegoForward and LegoSlideRight will move the robot as expected, based on the current orientation specified by LegoDir.

When the program in **Figure 10** executes, the robot slides gracefully along the wall, rather than turning as a more conventional robot might do. Experimenting with parameters such as motor speed and the distance to move toward or away from the wall can fine-tune the robot's behavior.

The complete source code for the library, as well as a link to a YouTube video demonstrating the robot in action can be found at **www.RobotBASIC.com**. Use our library or adapt the principles discussed here to create your own. Either way, we think you might enjoy experimenting with unconventional wheel systems. Who knows ... you might find them perfect for your next robotic project. **SV**

Discuss this article in the *SERVO Magazine* forums at **http://forum.servomagazine.com**.

Extending The Beginner Bot With The PropBOE Part 5

by Gordon McComb

Discuss this article in the *SERVO Magazine* forums at **http://forum.servomagazine.com**.

You're only a beginner once. As your skills evolve, you transform from beginner to novice, then to apprentice, and finally expert. It's a gradual process, and in the world of robotics, a potentially expensive one as you collect new hardware to play with.

The Beginner Bot expandable platform is one way to ease the burden of starting from scratch each time you cut a new notch in your robotics belt. The idea is that you start with a small, inexpensive (under $20) chassis, then swap out old parts and add new ones as you progress through your robot building career.

The last four installments of this series have covered how to build the Beginner Bot platform using wood or plastic, how to steer it using mechanical switches from a tethered control panel, and how to convert the robot to fully electronic control — complete with basis sensors.

Previous articles have also described adapting the Beginner Bot to the popular PICAXE and Arduino microcontrollers. In this fifth and final part, you'll see how to use the Parallax Propeller to provide fully autonomous control.

The finished Beginner Bot as described in this article is shown in **Figure 1**. Before continuing, note that this article relies on construction details described in the earlier parts of this series. There's no need to build each stage of the Beginner Bot, but if you're just starting out you'll want to refer to the earlier articles.

All About the Parallax Propeller

Like the PICAXE and Arduino — we covered these in earlier articles in this series — the Parallax Propeller is a programmable microcontroller that can be used to interface with external devices. Common applications include running motors, reading the status of switches, checking temperature probes ... you name it.

What sets the Propeller apart from most

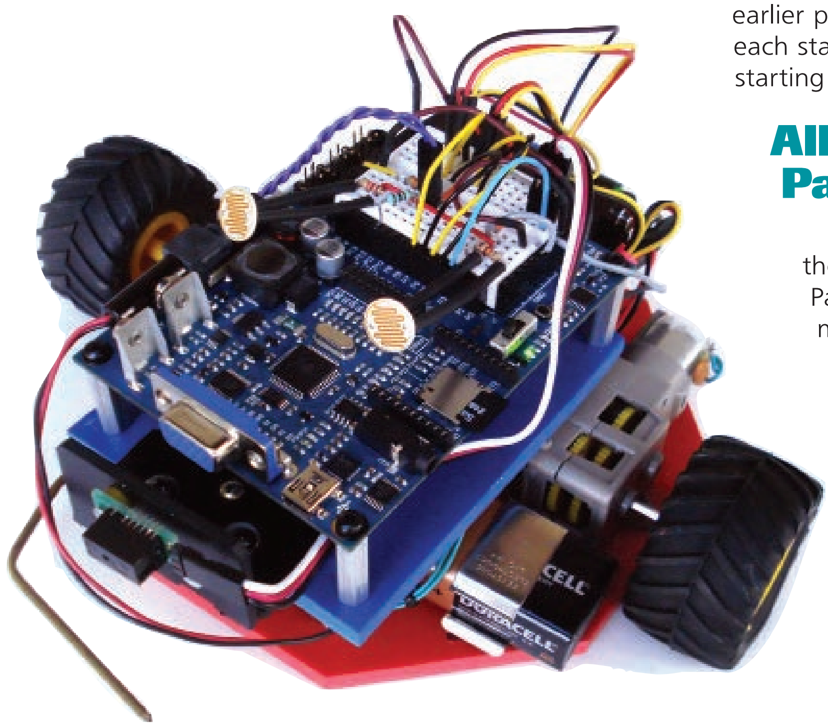


FIGURE 1. The completed Phase 5 version of the Beginner Bot with a Parallax Propeller Board of Education (PropBOE) microcontroller development board, complete with sensors and wiring.

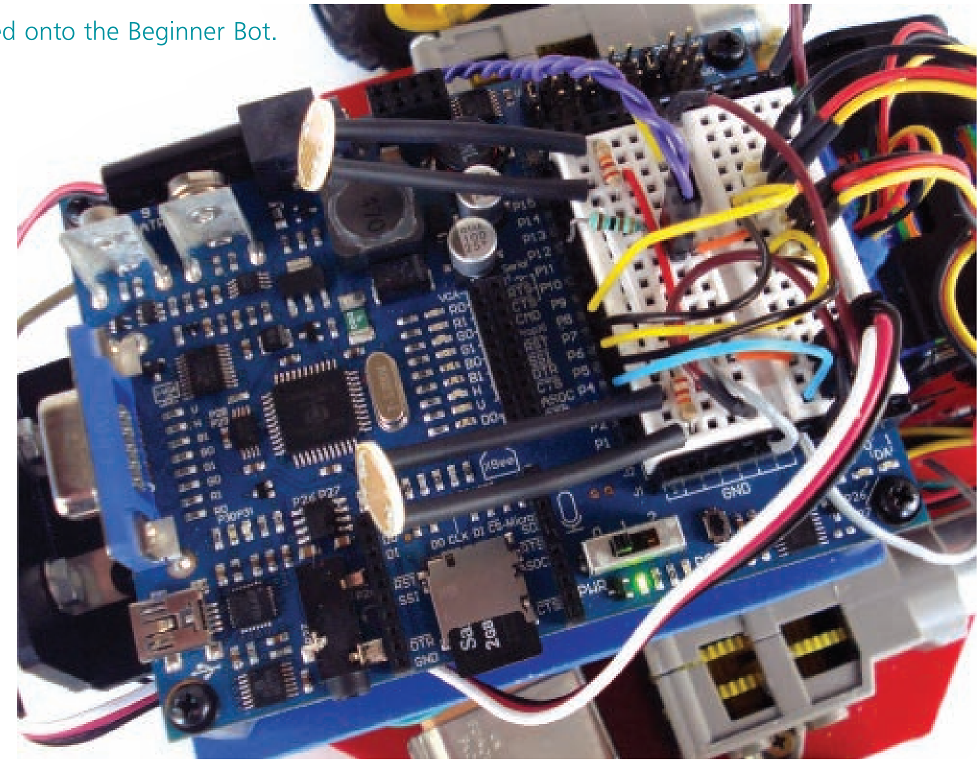


FIGURE 2. The PropBOE, mounted onto the Beginner Bot.

other microcontrollers is that it's designed from the ground up to be multi-tasking. Inside one Propeller is actually eight individual microcontrollers, each operating independently but able to cooperatively share resources like memory and input/output pins. You don't have to use the multi-tasking feature of the Propeller (none of the simple demonstrator programs in this article require it), but it's nice to know the ability is there, should you need it.

The chip uses eight general-purpose cores, or "cogs," to do most of the heavy lifting required of the typical microcontroller. This makes the Propeller a bit different than the typical microcontroller. For example, instead of relying on hardware timers — special circuit blocks inside the chip that do just one job — in the Propeller these types of tasks are handled by one or more simultaneously running cogs.

The Propeller is supported by numerous programming languages, including Basic and C, but the two primary languages used with the chip are unique to it: Spin and Propeller Assembly. Feel free to skip Assembly for now; most of what you'll want is easily handled by Spin. Parallax supports a large library of premade objects that provide common functionality which further decreases the code writing you need to do. You can learn Spin using the free documentation provided on the Parallax website. Be sure to check out Jon Williams' *Spin Zone* column in *Nuts & Volts* — the sister publication to *SERVO Magazine*.

The Propeller is just an IC, available in both surface-mount and 40-pin DIP. While you can construct circuits with a bare Propeller chip, more often you'll want to use a pre-made development or prototyping board which provides a voltage regulator to supply the required 3.3 volts to the Propeller, USB connection for programming from a computer, plug-in headers for wiring components to the Propeller, and various other support electronics.

There are a number of Propeller-based development and prototyping boards available, both from Parallax and from third party sources. For the Beginner Bot, I'm using a Propeller Board of Education, or *PropBOE*. It's a relatively new product — in fact, it was in its last development stage when I wrote this article. As a development board, it's a bit more expensive than some but its feature set is remarkable, providing for easy upgrades and expansion. For the Beginner Bot, I'll only be using a fraction of what the PropBOE has to offer, but you're free to go further on your own.

The main points of interest on the PropBOE include:

- *Integrated USB adapter.* Connect the PropBOE to your Windows-based PC for programming using a USB cable. Before first use, be sure to download the Propeller software package which includes the necessary USB driver to connect your computer to the PropBOE.
- *Dual voltage regulators*, for both 3.3 and five volts. Power supply connections include nine-volt battery terminals and a standard 2.1 mm barrel plug. The supply voltage should be in the range of six to nine volts.
- *Solderless breadboard area* with connection points to power, as well as the first 16 I/O pins of the Propeller.
- *Headers for connecting up to six R/C servos*. We won't be using this feature in the Beginner Bot, but it's handy to know they're available.
- *Integrated micro-SD data card* for reading and storing data in permanent and replaceable memory.
- *Plug-in headers for wireless modules*, including standard XBee transceivers.
- *Audio and video output terminals*. Video out is through a standard 15-pin VGA connection; audio is through a 1/8" stereo jack.
- *Built-in analog-to-digital converter*, for sampling the voltage level of several analog sensors at once.

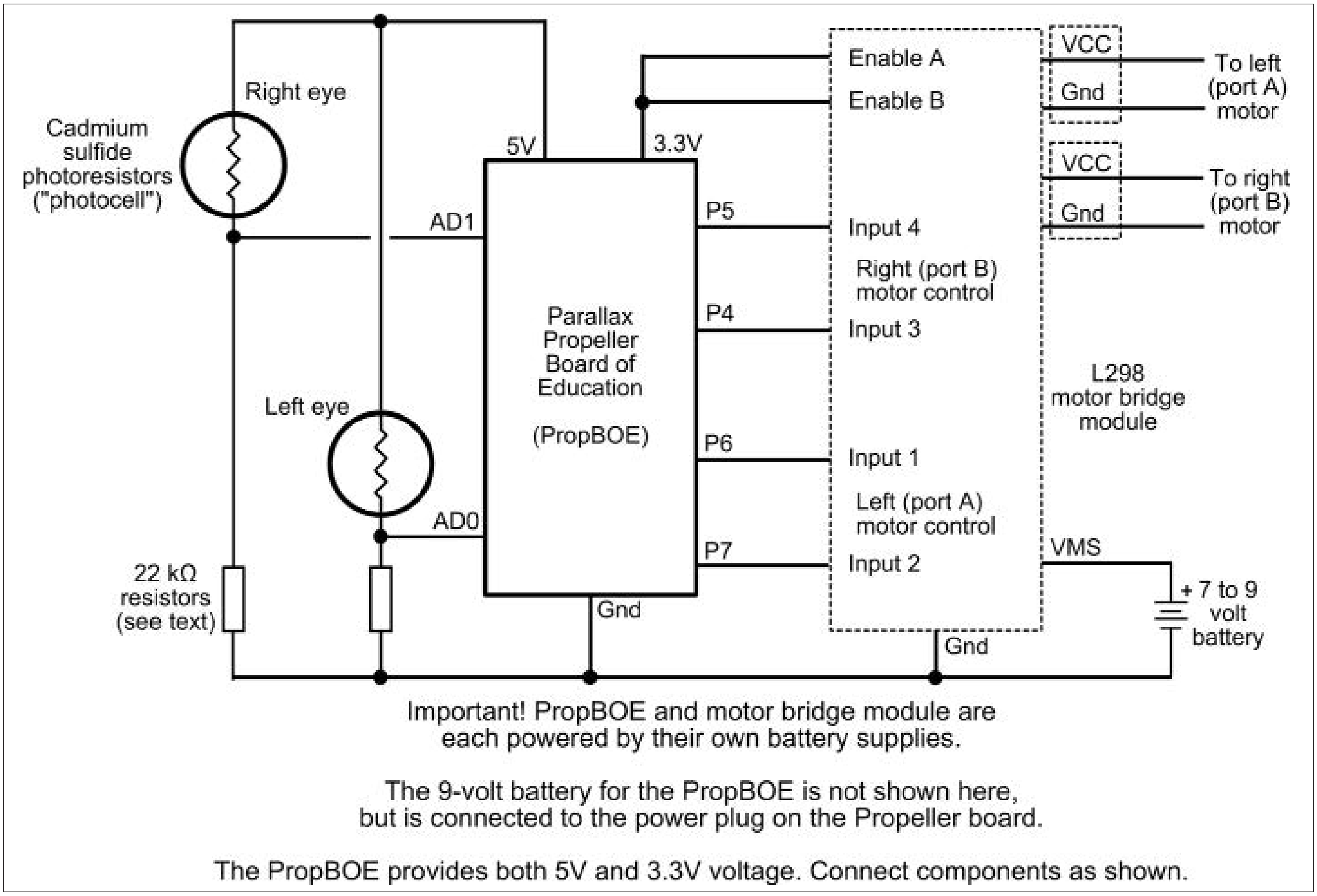


FIGURE 3. Schematic view of the CdS photocells, PropBOE, and L298 motor driver board. The PropBOE is operated from its own nine volt battery, connected to the barrel power plug.

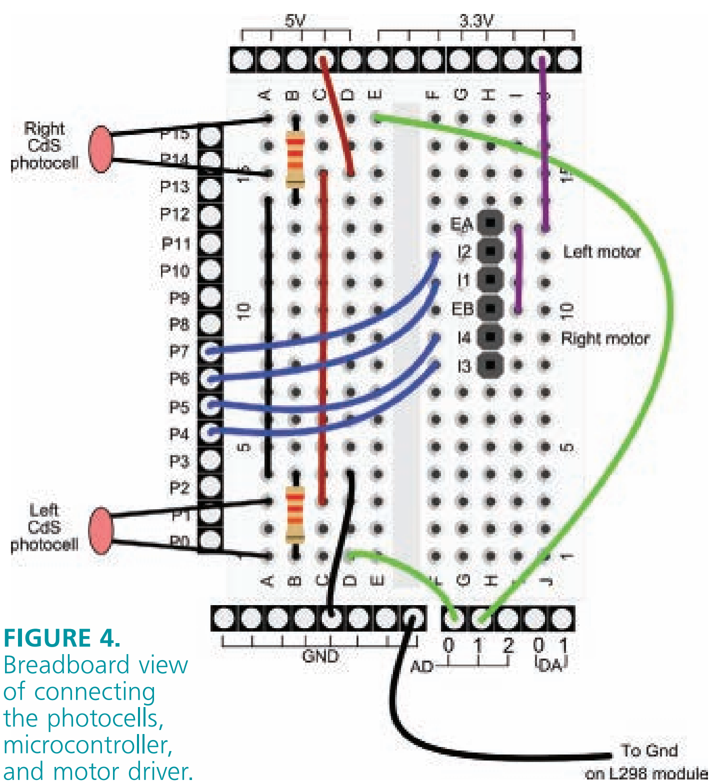


FIGURE 4. Breadboard view of connecting the photocells, microcontroller, and motor driver.

Mounting the PropBOE

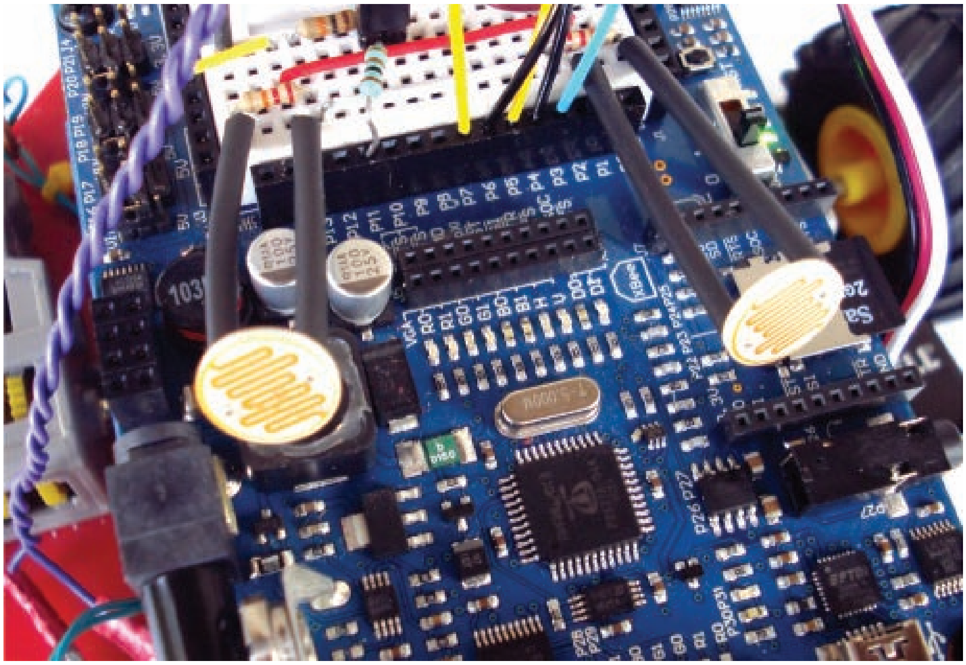
The PropBOE measures 3" by 4" inches. You'll want to securely attach it to the Beginner Bot (just like in **Figure 2**), using a mounting deck with hardware standoffs.

The PropBOE is just a tad bigger than the second deck we've used in previous installments for mounting the PICAXE and Arduino microcontroller boards (see Part 2 of this series for dimensions and mounting details). Construct a new mounting deck that's about 3-1/4" by 4-1/4" with mounting holes in the four corners that match those in the PropBOE. Use 1/2" (or longer) threaded spacers to act as standoffs between the mounting deck and the PropBOE.

For the deck material, you can use the switch control panel described in Part 1 of the series, or cut a new piece of plastic or wood to size. I opted for the latter, but feel free to adapt parts from an earlier phase of the Beginner Bot to save money. You can keep the control panel with its original dimensions, or cut it down to size.

By using longer standoffs, you can tuck extra components underneath the PropBOE — just be sure

FIGURE 5. Cadmium sulfide photocells attached to the solderless breadboard area of the PropBOE.



not to short-circuit anything! For example, you might want to place a small amplifier and speaker underneath, so you can add sound, music, or voice to your robot. Keep in mind that the balance skid of the Beginner Bot is placed in the rear, under the main batteries. That's where most of the weight is. Avoid adding too much weight to the front of the robot, or it won't ride on the rear skid. Of course, you can always move the skid from the rear to the front if your bot tips the wrong way.

Wiring for Motor Control

In the first part of this series, you learned how to use switches for manual control of the bot's motors. You then learned how to convert to electronic control using an H-bridge module, then adapt the electronics for use with a microcontroller.

Refer to Part 2 on how to mount the H-bridge to the Beginner Bot, connect the motors to it, and wire a battery pack to it. I recommend using a six-cell AA battery holder and rechargeable nickel-metal hydride (NiMH) batteries. Keep the batteries freshly charged, as a lower than normal supply voltage can cause mysterious and hard-to-track problems.

Refer to **Figure 3** for how to connect the PropBOE to the H-bridge module, including a set of cadmium sulfide (CdS) photocells that act as simple eyes. Use the mini breadboard and header connections as the interface between the motor bridge, the two photocells, and the PropBOE. **Figure 4** shows the circuit in breadboard view.

Note that the PropBOE gets its power from its own nine volt battery. A common ground connection is used between the PropBOE and the H-bridge module. This is required for proper operation. If you leave off the ground between the two circuits, the motors on your Beginner Bot may not work or they may function erratically. (Unlike that shown in Parts 2 and 3 of this series, the PropBOE does *not* require 5V power from the motor driver board.)

Use a piece of Velcro™ to secure the nine volt battery to the bottom deck. There's room on the left side, in front of the motor. Make or purchase a battery cable that has the standard nine volt battery clip on one end and a 2.1 mm (center positive) barrel plug on the other. The cable should be about 6" to 8" long.

The basic operating circuit is straightforward: Two photocells detect the amount of light falling on them. The photocell exhibits a change of resistance, depending on the amount of light. The less light, the higher the resistance; the more light, the lower the resistance. For each CdS "eye," a 22 kΩ resistor turns the resistive output to a varying voltage. The resistance of the CdS cell plus the fixed resistor form a voltage divider circuit.

Note that the photocells are connected to the 5V supply, rather than 3.3V. This is intentional, in order to increase the sensitivity of the readings. While the Propeller is powered at 3.3V, the analog-to-digital converter circuit on the PropBOE is run at 5V so it can be interfaced to circuits that operate at up to five volts.

Attach the photocells to the mini solderless breadboard as shown in **Figure 5**. Gently bend the leads of the cells so that they point slightly upward and outward. Add heat shrink tubing (unshrunk) over the photocell leads to provide both mechanical support and electrical insulation.

Use the Right Motors!

The Beginner Bot uses a pair of Tamiya gearboxes that have been modified according to instructions provided in Part 2 of this series. Specifically, the motors used in the gearboxes have been replaced with versions that provide for operation at six to 12 volts, and with higher efficiency. These motors are available from Pololu (item #1117), among other sources. Cost is under $2 each.

Be sure to *not* use the stock motors that come with the Tamiya gearboxes. These are rated for only three volts and can consume copious amounts of current. This current exceeds the rating of the L298 H-bridge used to control the motors.

LISTING 1 – !RobotTest.spin

```
OBJ

  pin  : "Input Output Pins"
  time : "Timing"

PUB BotTest

  repeat
    RobotFwd
    time.Pause(2000)
    RobotRev
    time.Pause(2000)
    RobotRight
    time.Pause(2000)
    RobotLeft
    time.Pause(2000)
    RobotStop
    time.Pause(2000)

' motion routines
PRI RobotFwd
    pin.Low(4)
    pin.High(5)
    pin.High(6)
    pin.Low(7)

PRI RobotRight
    pin.High(4)
    pin.Low(5)
    pin.High(6)
    pin.Low(7)

PRI RobotLeft
    pin.Low(4)
    pin.High(5)
    pin.Low(6)
    pin.High(7)

PRI RobotRev
    pin.High(4)
    pin.Low(5)
    pin.Low(6)
    pin.High(7)

PRI RobotStop
    pin.Low(4)
    pin.Low(5)
    pin.Low(6)
    pin.Low(7)
```

LISTING 2 – !LightSeek.spin

```
OBJ

  system : "Propeller Board of Education"
  adc    : "PropBOE ADC"
  pin    : "Input Output Pins"
  time   : "Timing"

VAR

  long ad0
  long ad1
  long thresh

PUB Go

  system.Clock(80_000_000)
  thresh := 900

  repeat
    ad0  := adc.In(0)
    ad1  := adc.In(1)
    if ad0 > thresh AND ad1 > thresh
      RobotFwd
    elseif ad0 > thresh AND ad1 < thresh
      RobotLeft
    elseif ad0 < thresh AND ad1 > thresh
      RobotRight
    else
      RobotStop
    time.Pause(5)

[duplicate "motion routines" from Listing 1 here]
```

TABLE 1

Input A	Input B	What Happens
Low	Low	Motor stops
Low	High	Motor turns one direction
High	Low	Motor turns the other direction
High	High	Motor stops

The voltage produced by the CdS sensors stretches from between zero and five volts, and is connected to two of the PropBOE's analog inputs — pins marked AD0 and AD1 (these are located near the bottom left of the board). The value of 22 kΩ for the resistors connected to each CdS cell is determined by trial and error. You may want to try different values to find the best sensitivity for the photocells you're using. You want the highest sensitivity while maintaining the widest possible swing between zero and five volts.

As you read in Part 2 of the Beginner Bot series, the Seeedstudio L298 H-bridge module requires at least two inputs per motor. The direction of the motor is determined by the instantaneous value of these two inputs, according to **Table 1**.

You control the operation and direction of either motor by setting the pins LOW (zero volts) or HIGH (3.3 volts). You'll see exactly how this is done next.

Testing the Motors for Proper Operation

As you develop your robots, you always want to test each new feature or component you add to them. Refer to **Listing 1** for a demonstration program for checking the basic operation of the PropBOE, the H-bridge, and the motors.

Download this program and its additional library from the *SERVO* website into the Propeller Tool development environment, then:

1. Place a small block under the Beginner Bot base to lift the wheels off your worktable.
2. Connect the battery to apply power to the H-bridge, and the nine volt battery to the PropBOE.
3. Plug in the programming cable between your PC and the PropBOE. Start the Propeller Tool IDE program (included in the Propeller software download), and choose the *Run->Identify Hardware* command (or press F7). The software

should detect the PropBOE connected to your computer. If it doesn't, check the connection and be sure power is applied to the PropBOE. Its green power indicator should glow.

4. Compile the program by choosing *Run->Compile Current->Load EEPROM* (or press F11).

Important! The program in **Listing 1** requires the use of several *object libraries* which add important functionality. These libraries — provided by Parallax — are included with the program download on the *SERVO* website. If any of these object libraries are missing, the program will not compile or run. Also take note of the indentation used in Spin programs. It's important. The indenting is how Spin keeps track of program blocks, like *if* conditional tests and repeating loops.

The program is automatically compiled and uploaded to the PropBOE. Upon successful compilation, the program runs automatically. Assuming everything has been connected properly, the motors should turn in various directions as the robot goes through its motion routines.

Fully test the robot by first disconnecting the USB cable from the PropBOE and temporarily unplugging the nine volt battery. Place the robot on flat ground and plug the nine volt battery back in. The robot should go through its motion routines: forward, backward, turn right, turn left, and stop. If one or both motors turn in the wrong direction, remove power and flip the terminal wiring from the affected motor on the H-bridge.

The best surfaces for testing are tile, wood, or a kitchen table. Carpet is acceptable as long as it has a very low nap. If the bot appears to struggle as it's moving along, relocate it to ride over a smoother surface.

Using Light to Control Your Robot

Listing 2 shows how to control the Beginner Bot using a flashlight by shining the light into the photocell eyes. *As with the previous program example, **Listing 2** also makes use of object libraries. You need these files — which are listed at the top of the program — in order for the project to compile.*

The program tells the microcontroller to read the value from both photocells. A series of *if* conditional logic tests determine if there's enough light to follow, and if so, in what direction the robot should travel.

LISTING 3 – !ReadADC.spin

```
OBJ

  system : "Propeller Board of Education"
  pst    : "Parallax Serial Terminal Plus"
  adc    : "PropBOE ADC"
  time   : "Timing"

VAR

  long ad0
  long ad1

PUB Go

  system.Clock(80_000_000)

  repeat
    ad0 := adc.In(0)
    ad1 := adc.In(1)
    Display
    time.Pause(50)

PUB Display

  pst.Home
  pst.Str(String("ad0="))
  pst.Dec(ad0)
  pst.Str(String(", ad1="))
  pst.Dec(ad1)
  pst.ClearEnd
```

Note the *threshold* value which is used to set the boundary between dark and light. I've set the light/dark threshold to 900 — out of a range of 0-1023 — as a starting point. Try higher or lower values to see what works best with your particular CdS cells.

When both cells receive light over the threshold, the robot drives forward. When only one cell receives light over the threshold, the robot turns in the direction of the light. If neither cell receives light over the threshold, the robot stops.

Upload the program in **Listing 2**. Remove the programming cable when transfer is complete. Move to

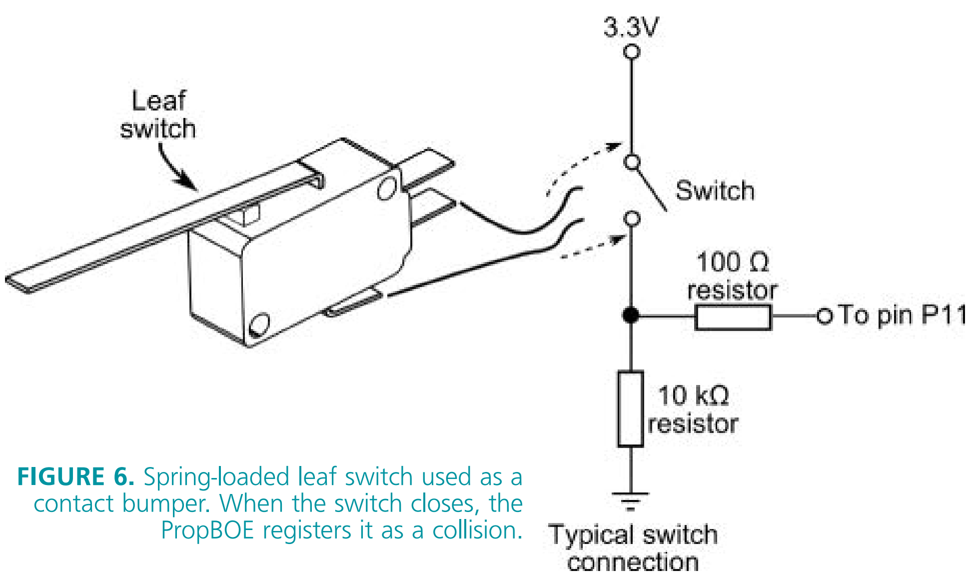


FIGURE 6. Spring-loaded leaf switch used as a contact bumper. When the switch closes, the PropBOE registers it as a collision.

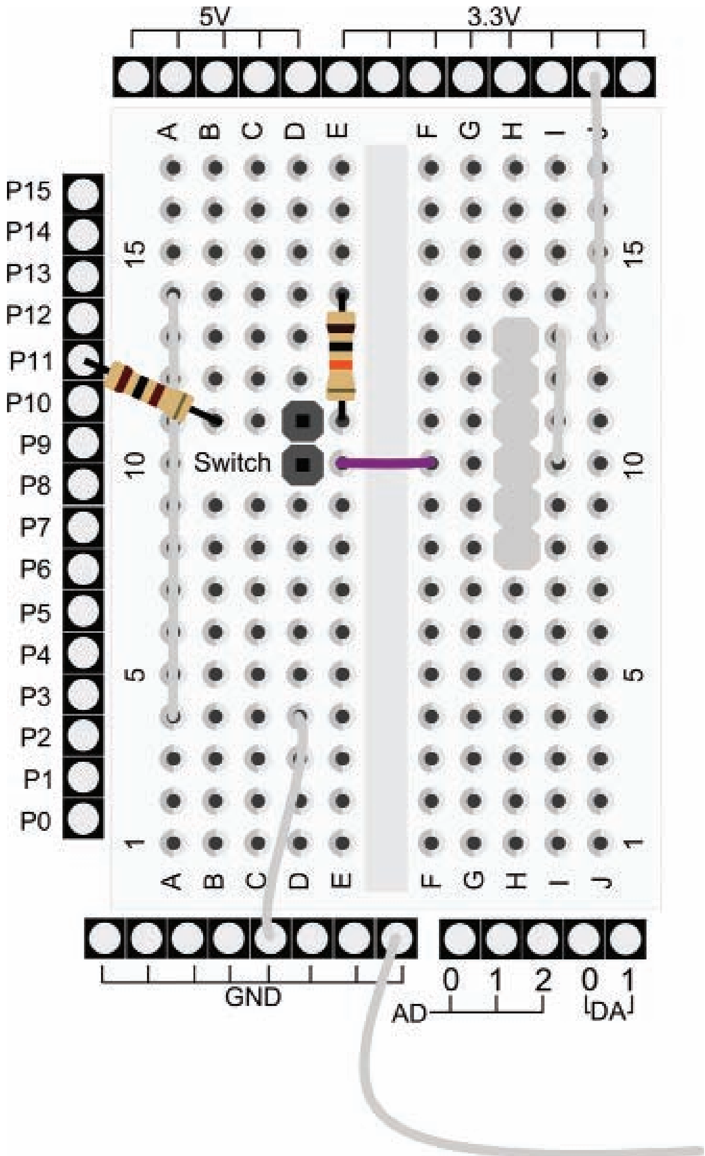


FIGURE 7. Breadboard view of attaching the wires from the switch to the PropBOE.

Sources

Precut and predrilled Beginner Bot base, with all construction hardware:

Budget Robotics
www.budgetrobotics.com

Propeller Board of Education (PropBOE), jumper wires, etc.:

Parallax
www.parallax.com

Seeedstudio L298 motor bridge module (see Part 2 of this series for details), additional components:

HVW Tech
www.hvwtech.com

Jameco Electronics
www.jameco.com

Mouser Electronics
www.mouser.com

Pololu
www.pololu.com

RobotShop
www.robotshop.com

Solarbotics
www.solarbotics.com

a darkened room, apply power to the robot, and place it on the ground, away from any light. The robot should not move. Next, aim the flashlight evenly onto both photocells. The robot should move toward the light. Aim the flashlight into just one photocell. The robot should turn toward that photocell.

Does your robot move when there's no light falling on the CdS cells? Try changing the threshold value. Conversely, if the light from the flashlight seems to make no difference, enter a lower threshold and try a darker room.

Use the program in **Listing 3** to help determine the threshold value. Compile and upload the program to the PropBOE. Keep the USB cable connected between your PC and PropBOE. Open the Parallax serial terminal program included with the Propeller software package. (If the terminal program doesn't automatically detect the serial port in use by the PropBOE, you may need to manually select it. The baud rate should be set to 115200).

Shine a flashlight into the photocells. The values of both cells should be shown in the terminal. Note a safe light/dark threshold when the photocells are in ambient (natural) room light.

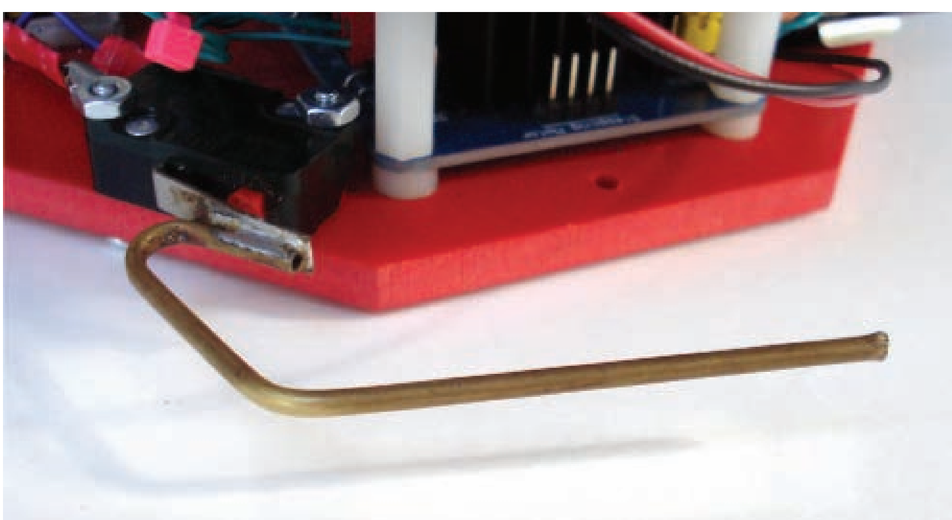


FIGURE 8. Mount the switch on one side of the front of the Beginner Bot. By attaching a stiff wire to the leaf (solder or glue), you can extend its reach across the entire front of the robot.

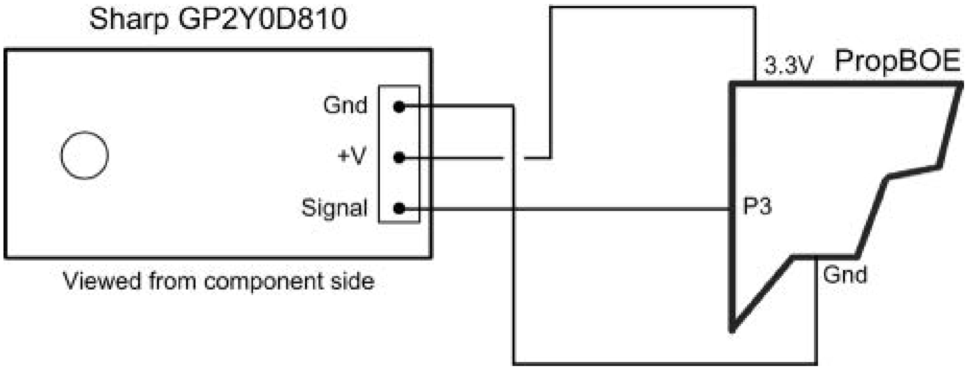


FIGURE 9. Wiring diagram for connecting the Sharp GP2Y0D810. Be sure to observe the correct polarity of the 3.3V and ground connections.

Sensors for Object Detection

One of the simplest and least expensive ways to look for objects in the path of the robot is the leaf switch, shown in **Figure 6**. When the switch closes, the circuit is completed and the PropBOE registers it as a contact with an object.

Refer to **Figure 7** on how to connect the switch to the PropBOE. I like to use leaf switches as bumpers because they already have a fairly large area of contact. You may even wish to extend the bumper of the switch by soldering or gluing a piece of coat hanger wire to the leaf, similar to **Figure 8**.

Wiring the switch to the PropBOE requires a couple of extra resistors which are shown in the diagrams. The 10 kΩ resistor acts as a pulldown and provides a consistent and reliable LOW value when the switch is inactive. When the switch is closed, the output goes HIGH. In keeping with the how-to circuit examples provided by Parallax, the 100 ohm resistor provides current limiting protection to the Propeller input.

Another relatively inexpensive sensor is the digital infrared proximity detector, specifically a Sharp GP2Y0D810 (available at Pololu; see the **Sources** box). Attach a three-wire servo extension cable between the sensor and the Beginner Bot's breadboard. (Be sure to get some extra double-length male headers while you're at it.)

I've connected the sensor for my Beginner Bot to a homebrew mounting bracket. I made the bracket so that it could be swiveled out of the way when I need to plug the USB cable into the PropBOE. Refer to the wiring diagram in **Figure 9** for the basic connection, and the breadboard view in **Figure 10**.

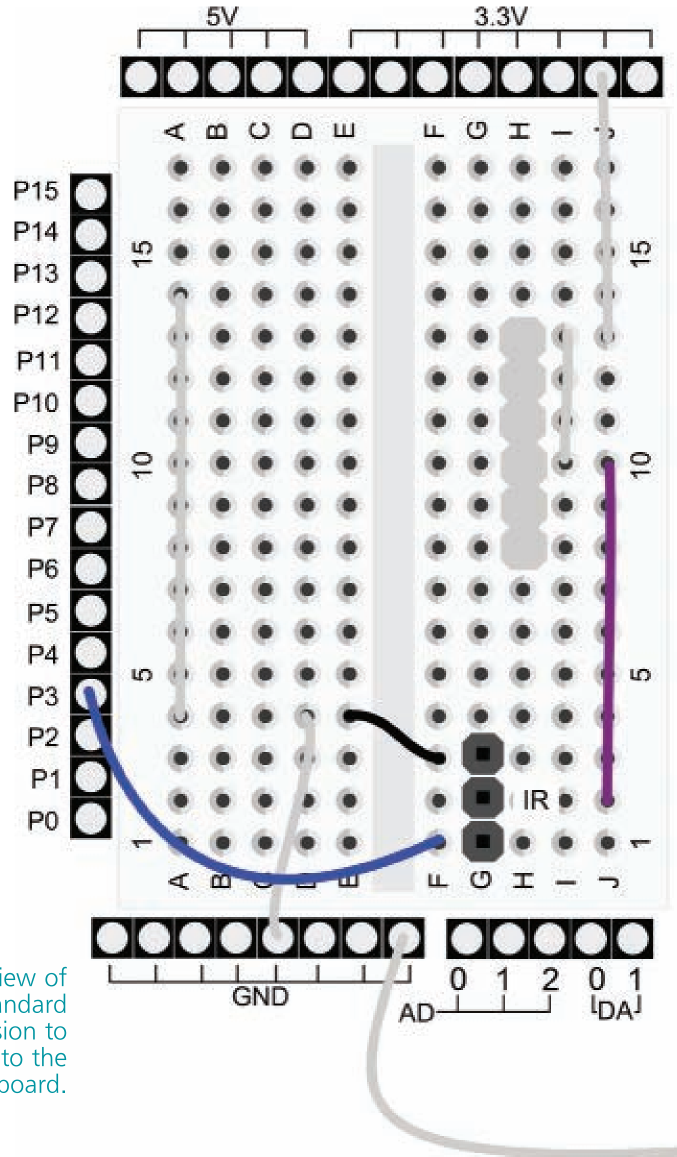


FIGURE 10. Breadboard view of the GP2Y0D810. Use a standard three-wire servo extension to connect the sensor to the breadboard.

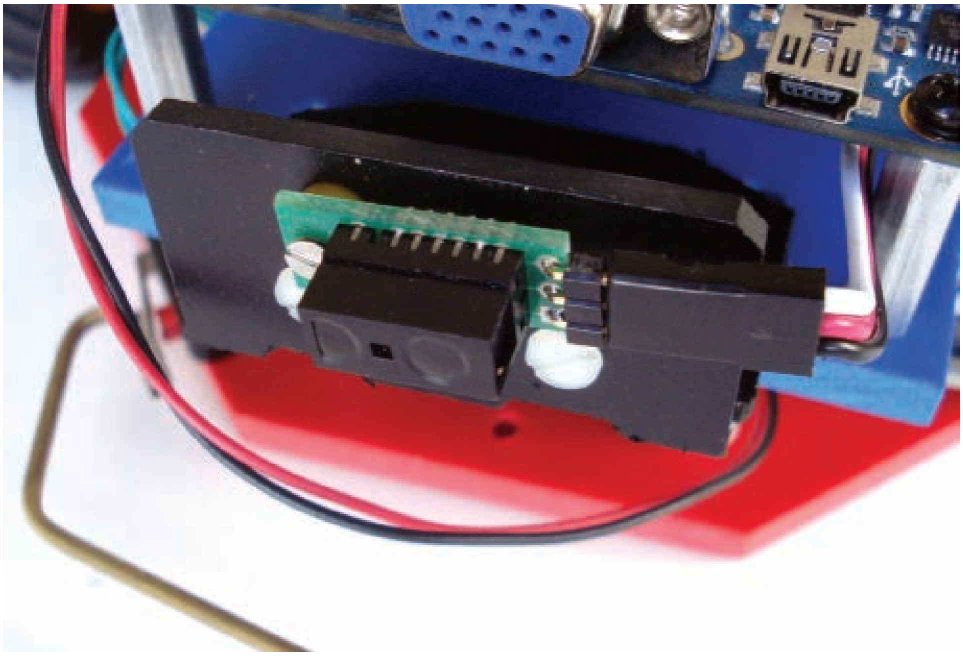


FIGURE 11. The GP2Y0D810 infrared distance judgement sensor, mounted on the front of the Beginner Bot on a homemade bracket.

The mounted IR detector is shown in **Figure 11**.

See **Listing 4** for a demonstration of using the Beginner Bot with the switch and infrared object detection sensors. This program — like the rest — requires the use of separate object library files. They're included with the file download from the *SERVO* website.

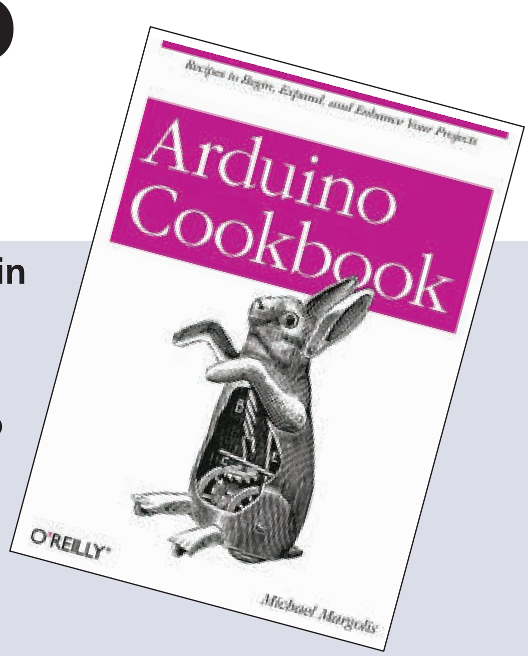
The program first reads the current output of the bumper switch. If it's HIGH, the switch has been triggered. The robot reacts by momentarily backing up, turning to the right, and then proceeding forward again.

Next comes a test to see if the IR detector has been activated. If it's LOW, there's an object within its field of view

Book Review

by Steven Nelson

The Arduino Cookbook



I had just got back from the 2011 Maker Faire in San Mateo, CA., and was excited about the huge bunch of cool machines running with the Arduino microcontroller. So, of course, I had to purchase one. I bought the Arduino Mega 2560. I had never used an Arduino before, but was familiar with other microcontrollers such as the Parallax BASIC Stamp 2 and the Propeller chip.

which extends out to about 10 centimeters in front of the sensor. If triggered, the robot backs up briefly, turns to the left, and then continues.

Going Further With the Beginner Bot

So ends our journey with the Beginner Bot. With it, you learned basic robotic control principles, how to operate motors electronically, using light and touch to detect objects, and ways to adapt the robot to three different microcontrollers.

This is by no means all that you can do with your Beginner Bot. Feel free to experiment on your own. All of the variations of the Beginner Bot use a solderless breadboard for making easy connections to electronic components. To try something different, you only need to yank out the old circuit and plug in the new. **SV**

LISTING 4 – !RobotWander.spin

```
OBJ

  pin  : "Input Output Pins"
  time : "Timing"

PUB BotWander

  repeat
    RobotFwd

    if pin.In(11) == 1
      RobotRev
      time.Pause(1000)
      RobotRight
      time.Pause(1000)

    if pin.In(3) == 0
      RobotRev
      time.Pause(1000)
      RobotLeft
      time.Pause(1000)

    time.Pause(50)

[duplicate "motion routines" from Listing 1 here]
```

www.nutsvolts.com/index.php?/magazine/article/december2011_Nelson

When my new electronic treasure arrived, we sat side by side, searching the Internet for tutorials and information on how to communicate with each other. We found the Arduino website which is a very useful and excellent resource. What I found while using the computer though, is it is often a bit clumsy trying to read the text on the computer that's explaining the software you're trying to write while you're trying to write it. This left me longing for a printed textbook.

During a trip to the local bookstore, I found another treasure: the *Arduino Cookbook*. This book covers the basics of setting up your computer to work with the Arduino, and then moves on into programming and circuit design using this amazing microcontroller. There are 18 chapters in the book, each filled with all kinds of tips and little nuggets of wisdom. Every time I open this book, there is something new and exciting to explore.

For the robot builder, there are many good topics discussed such as reading, switches, sensors, and Internet or wireless communication devices. It also covers controlling LEDs, sending PWM (pulse width modulation) signals to servos, and driving high-powered motors with an H-bridge or even a speed control. The nice thing is the book gives you all the code, plus the electronic schematics which are well documented. All the software examples are available for download online at the publisher's website.

Switching over to the Arduino from the BASIC Stamp 2 had me a little concerned about how difficult it would be to understand this new chip and programing language. Well, after finding this book it appears this will be an easy transition and a lot of fun. I would rank it five out of five stars for a reference book. **SV**

Web Sources

The *Arduino Cookbook*
http://oreilly.com/catalog/9780596802486

The Arduino Home Page
www.arduino.cc

The Arduino Cookbook by Michael Margolis
Copyright 2011
O' Reilly Media, Incorporated

by Matt Bunting

Pimp My Hexapod

Part 1

In the September '11 issue of *SERVO*, the Geerhead column was about my hexapod design and implemented projects. The platform itself has been an evolutionary, iterative design where I am constantly upgrading parts. Since it's been been a year and a half since the last major upgrade, plans for a better system are now designed to alleviate the most common issues.

I have been fascinated with robotics nearly since I can remember. There was always something so intriguing to me about building a small autonomous machine that could interact with the world. The idea of building a robot was akin to building a little creature. It has always generated philosophical ideas, making me think about how I accomplish a very mediocre task, and then how to mimic that mathematically, electronically, and mechanically. Very quickly it became apparent that even some of the simplest tasks are nearly impossible to accomplish with our current technology.

As a child, I had built many rover robots using LEGOS, radio-controlled cars, and wireless cameras. The use of a hot glue gun was key to much of my success. Though none of my early robots were autonomous, I always wanted to build an even more complex, elegant machine. When I first saw a hexapod, the extraordinary number of motors immediately piqued my interest. I thought to myself that I would one day build a hexapod.

During high school, I dabbled here and there trying to build a hexapod by scrounging up all the motors I had, but I could not quite get to the point of making a strong enough hexapod to walk. After working for a couple years and saving as much money as I could, I was able to splurge on some better motors. I was able to upgrade my original design into a working, walking hexapod. I also had a great excuse at the time, as I needed to build a robot for a Cognitive Robotics class.

I resurrected what I had previously built and added some better components, but it was nothing more than an upgrade of my original build. The project turned out to be pretty successful; I had the robot learn how to walk by implementing Q-Learning, even though it took four straight days for it to finally get it.

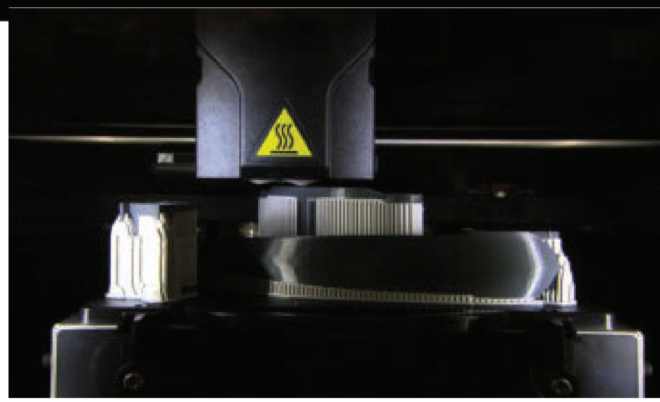



FIGURE 1. Inside the Dimension 3D printer as it finishes up a femur, tibia, and coxa.

Original Setup

The following summer, I began work at the Robotics and Neural Systems Laboratory (RNSL) under professor Dr. Anthony Lewis, who taught the course where I built the Q-Learning hexapod. Many of the robots designed in the lab were built using the Dimension BST 1200ES 3D printer created by Stratasys. Naturally, anticipation started to build as I began thinking of ways of upgrading the hexapod. Originally, I had brass pieces purchased from my local hardware store soldered together with mechanical components not being quite as symmetric as I had envisioned. I knew that a full robot makeover was in order, so I began doing some research of other upgrades to implement. **Figure 1** shows the 3D printer extruding some of the first hexapod leg components. **Figure 2** shows the progression from the original brass femur to the current split design femur.

The hexapod had been transformed from a Frankenstein mixture of parts to something very similar to what I have now. The Dynamixel series of motors from Robotis had been a clear decision for me since I purchased my first one six months prior to the upgrade. The idea of a tunable motor with error checking, temperature, and overload sensing with position feedback into the computer on an easy to wire daisy-chain communication bus was a dream for most hobby roboticists. Robotis made that a reality.

The total motor count consisted of six RX-10s, six RX-28s, and nine AX-12s. I did have to build a small circuit to communicate to both the TTL bus of the AX-12s and the RS-485 bus of the RX series, but after that little hiccup it is possible to communicate to the motors at 1 Mbps on the same bus. That implies the possibility of sending and receiving more than 100 read and write commands in the time it takes a regular servo motor to register one standard PWM cycle.

Compulab had just announced the release of the fit-PC2 just days prior to my decision to add an onboard computer for processor-intensive vision computations. The Atom z530 processor can easily crank out multithreaded code on a full Ubuntu install, while the whole computer only consumes five watts of power. It is possible to view the hexapod as basically a walking case mod.

The hexapod was my first computer to have Linux. There was a bit of a learning curve, and even after a few years I am still discovering what I can do, but even more importantly what *not* to do. I highly recommend using Linux, as I have been able to get very advanced behavior while maintaining low level control on a well supported, free operating system.

FIGURE 2. Various femurs produced from iterative design. The femur at the top left is the original brass femur. Over time, the femur has evolved to the split design on the lower right.

The camera I had previously used was a Logitech Quickcam Communicate Deluxe. Though a great camera that works seamlessly with Ubuntu and OpenCV, I grew tired

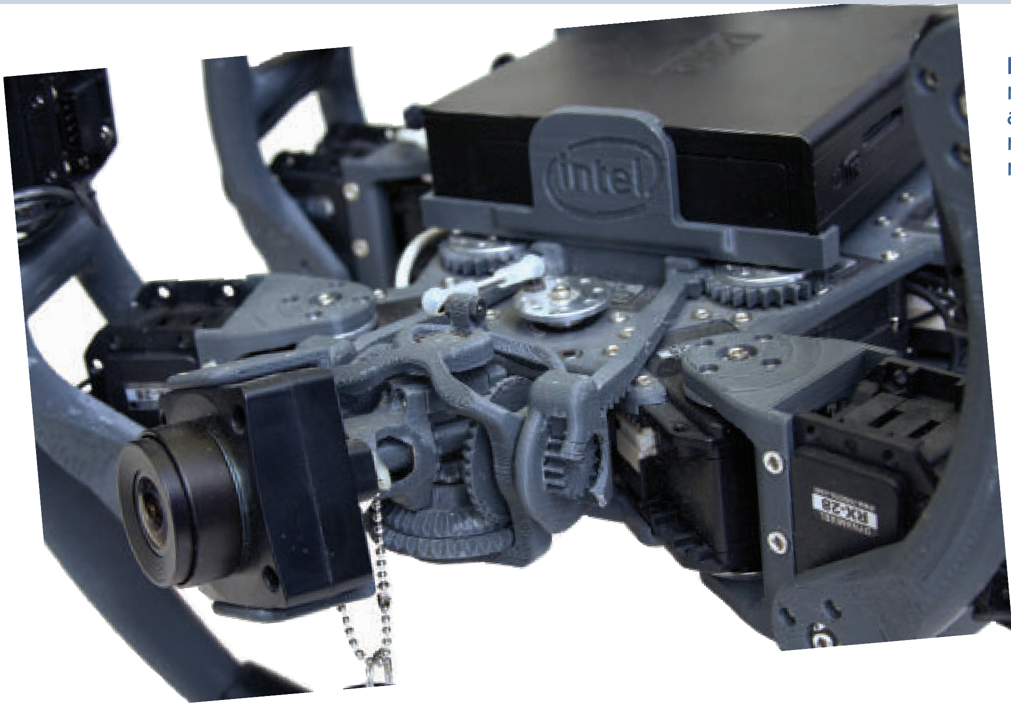


FIGURE 3. Close-up of the head mechanism. The use of bevel gears allows for panning, tilting, and rotation of the camera. The three motors sit just below the fit-PC2.

of the automatic image adjustment features that continually resulted in poor data for calculations based on successive images, including optic flow algorithms. I decided to invest in a machine vision camera, the Point Grey Chameleon. I highly recommend this camera mostly because it is a full machine vision camera: with some nice features, including HDR capabilities and a small GPIO port out the back. Pretty slick for a nice robust package that is even smaller than the Quickcam, assuming you can find a small enough lens. I was able to find a 150 degree field-of-view lens for $6 on eBay.

I designed a three degree--of-freedom neck mechanism based on a similar concept to a differential system. Not knowing proper gear design at the time, I made an attempt at designing and 3D printing spur and bevel gears.

Though the setup works, it doesn't do so very smoothly and takes a significant amount of break-in time. Three of the AX-12 motors listed above are hidden in the body and are used to operate the head mechanism. **Figure 3** shows more detail of the mechanism.

Current Setup

Once Intel discovered the video I had posted on YouTube, they decided to purchase a couple of my hexapods. I decided to do a small motor upgrade to 18 RX-28s and three DX-117s. The difference between the original and Intel models can be seen in **Figure 4**. The upgrade may seem overly powerful at first, but Intel had intended to use the hexapod all over the world at conferences and trade shows, running the motors 12 hours at a time at various temperatures. The Intel model uses a Playstation 3 Sixaxis controller to operate various parameters, from body tilt to walking to changing gaits. Using the controller, another mode can be enabled. The second mode implements face tracking. Here, the hexapod attempts to look at the closest person, turning and tilting according to their position.

The original setup was intended to be used as an in-house robot with occasional use, testing and implementing algorithms that stimulated my curiosity. A regular demo for a large company most certainly involves a different set of conditions that need to be accommodated for. Demoing a robot once a week for hours at a time in hot and cold temperatures can be much more harsh than one may realize. Even with the better motors, certain components have shown to not hold up under these strangely harsh conditions. Another upgrade is definitely in store for the hexapod.

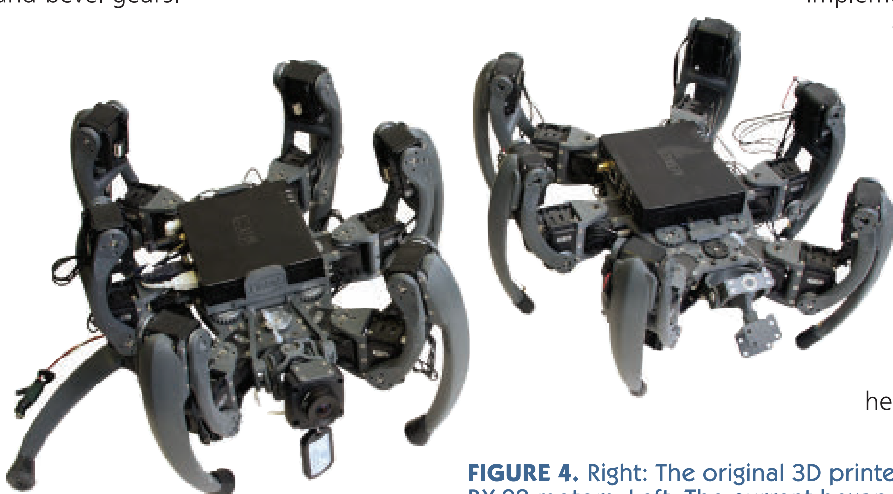


FIGURE 4. Right: The original 3D printed hexapod with AX-12, RX-10, and RX-28 motors. Left: The current hexapod with RX-28 and DX-117 motors. (The right middle leg is broken in the photo.)

Issues

Before upgrading and searching for better components, it is best to discover and list all the known issues with the current design. As with many things, the hexapod is kind of an iterative deign, always finding faults then upgrading the weakest components. Given that the last upgrade was nearly two years ago, the list has certainly grown fairly large.

FIGURE 5. The most common breaking points on the hexapod. The femur on the left has been patched using paperclips, business cards, super glue, and a water bottle wrapper. The coxa bracket on the right continually breaks at the servo horn screw holes.

1. Mechanical Components.

The use of the 3D printer has been nothing short of incredible. The hexapod was my first major project using SolidWorks and the 3D printer, and I would not have come as far as I have without the printer. The leg components are strong, inexpensive, can be produced hollow, and are machinable. The most important feature of the 3D printer is rapid iteration. It is possible to conceive an idea, model it in a CAD program, and have the part the same day. Most of the leg components of the hexapod were in my possession the same day I thought of the radical curved design. For research and hobby robotics, the 3D printer is an incredible asset.

When building a robot to run many hours, the plastic begins to show its weakness. Over time, the plastic begins to crack. When printing parts, it is best to consider where the strain will be, then print the part in the least likely way to crack. Typically, it is expected to crack along the layering direction or "along the grain," but repeated use and continuous pounding from stepping causes fractures to be perpendicular to the layers — or against the grain. These fractures can be seen in **Figure 5.**

LINKS

Personal Website:
www.12centdwarf.com

Robotis:
www.robotis.com

Intel:
http://edc.intel.com

Robot Source:
www.robotsource.org

Dimension 3D Printers:
www.dimensionprinting.com

Point Grey:
www.ptgrey.com

2. Head Mechanism.

Related to the above, some parts of the head mechanism break over time as they are synthesized from the same printer. As stated earlier, the gears have not been designed properly. Currently, each gear is simply a circular pattern of triangles around a thin cylinder. This results in a required breaking-in process so that the plastic is able to wear into a mating gear to mesh more effectively. Though they do perform transmission of torque as a normal gear should, the motion remains rough. Also, the smallest gears often break and teeth get sheared off. The head mechanism works okay for a little while, until more parts need to be printed and need to be broken in again. Being one of the more unique and interactive parts of the hexapod, something must be redone so that all demos involve a functioning head.

3. Motors.

I have loved using Robotis motors since I first powered up an AX-12. For the strength and features, I could not imagine switching to another type of motor. So far, I have used nearly every Dynamixel motor in various robot projects — anything from bipeds, quadrupeds, wheeled vehicles, etc. I have even built a clock using three AX-12 motors as a fun demonstration project. The EX-106 motor is scary powerful, exhibiting 106 kg-cm of torque.

One major component I have always found to be desired was a full PID controller. The tunable compliance methods implemented in the AX, RX, DX, and EX series

FIGURE 6. A nice little package of 18 Dynamixel MX-28 motors from Robotis.

are useful for specific control strategies. It is possible to make the motor operate stronger in one direction which is useful for a hexapod leg where it is mostly applying a force in a single direction. If set right, a person can physically move part of the robot without causing much strain on the motors. However, the simple linear or proportional feedback (P) nature of the onboard controller results in position errors or overshooting, and does not follow any standard control system strategy. If a derivative feedback term (D) were to be added, then the adjustable slope could be increased to help tighten up the error and increase the speed of the position adjustment. To polish things off, an integral feedback term (I) could be implemented to get rid of the error entirely.

The motors use a high quality potentiometer that enables continuous rotation. The potentiometers work by having a resistive material in a ring with a metal brushed wiper attached to the shaft. Long days of repeated motion in the same place causes the resistive material to wear. This is highly unusual, and I am inclined to believe that other factors such as hot temperatures helped induce wear. The resulting wear causes a huge misreading for feedback. The drastically changed value results in a violent shaking behavior of the motor. The only way to remedy the problem is to replace the potentiometer, which is not too bad using a screwdriver and soldering iron.

4. Software.

The more I write code, the better I become. Considering that the code currently operating the hexapod was written about two years ago, my skills have vastly improved. The current user interface for the demonstration mode with the Sixaxis controller is very limited. It is difficult to see what is going on when the fit-PC2 is cranking away at processing face tracking and kinematics algorithms. Also, the use of the Sixaxis controller is seamless but limited.

The face tracking algorithm was only able to effectively work at 11 frames per second at a down sampled image resolution of 160x120. This was the most effective I was able to have the hexapod track a face. The operation is not very robust as a person must be facing right at the camera with no tilt of their head in any direction. The face tracking mode is very difficult for most people to interact with.

FIGURE 7. All the motors neatly laid out for the next-generation hexapod.

Plans

Having listed out the known problems, I can proceed with an upgrade plan.

FIGURE 8. Testing involute gear design using the 3D printer and SolidWorks.

1. Mechanical Components.

There are two routes to upgrading the legs. I could attempt to beef up the legs in SolidWorks and continue to use the 3D printer plastic. The other option is to change materials entirely, such as aluminum. Considering the abuse the hexapod undergoes, switching to aluminum seems like the best option. The costs of getting CNC milled aluminum with the current shape are certainly much higher than printing plastic, but the reduction of downtime at demos should more than make up for the higher price tag. Some redesign will be in order because a 3D printer can easily create any shape, but a CNC mill takes knowledge of the machine's capabilities which must be accommodated for when designing.

2. Head Mechanism.

Ideally, a new head mechanism would be created with metal parts. Considering that metal parts have much less give compared to plastic, the parts cannot be worn into each other as well as before. In order to solve the problem, a properly designed gear must be fabricated. In other words, replace the triangles with proper pitch angle and circular involute curves for both the spur and bevel gears. Fortunately, there are many resources for designing proper, standard gears.

3. Motors.

Robotis recently announced and currently sells a new motor: the MX-28. There are some tremendous advantages at only $20 more than the RX-28. Mechanically, the motor is similar to an RX-28, but the onboard controller is drastically different.

a. Magnetic Angle Sensor: The potentiometer has been replaced with a magnet and a magnetic encoder: the AS5045. Not only does this increase the resolution from 1024 to 4096 and increase the effective servo operation over the whole 360 degree range, but it is also completely contact-less. This means incredible durability, and it will solve all shaking issues that some of the RX-28 motors were exhibiting.

b. PID Controller: The control system has been upgraded to a full PID controller, following a very commonly known control system. Being that all three gains are adjustable, the motor can truly be properly tuned for any system that can be modeled as a mass spring damper. Also, the gains can be set to zero, so not only is it a PID controller, but it could also be a P or PD controller, or something non-standard like PI, D, or ID. (I can't wait to start tuning these motors!)

c. Microcontroller: Switching from the Atmega8, the MX-28 now features an ARM Cortex M3. That is a lot of processing power for a single servo motor. This results in a communication bus increase from 1 Mbps to 3 Mbps. It is also apparently possible to install a Linux kernel on the Cortex M3.

4. Software.

There will be new features implemented in the hexapod program, including a better interface. Video feed will be streamed wirelessly to a server so that users can see how the face tracking is performing. The additional features will, however, take more computation, so considering that performance is also to be increased, a newer computer will be in order.

Rather than use the z530 Intel Atom processor, I will be implementing an E6XX series Atom processor. This newer processor has onboard video encoding and decoding, so vision processing and streaming should operate very smoothly. Intel has developed a development board named devboard which will mate nicely with a fairly low profile Q7 platform, available from iWave.

Next Time

Some parts such as new motors are starting to arrive, and I am busy cranking away at redesigning the next generation of hexapod. You can see the shipment of motors I just received in **Figures 6** and **7. Figure 8** shows properly designed spur and bevel gears using the involute method. I will have an in-depth review of the MX-28 motors, fabricate aluminum parts, describe the design of involute gears, and recode and port my system to a newer, faster platform in upcoming installments. **SV**

PR Lite — Build Your Own PR2

Running ROS

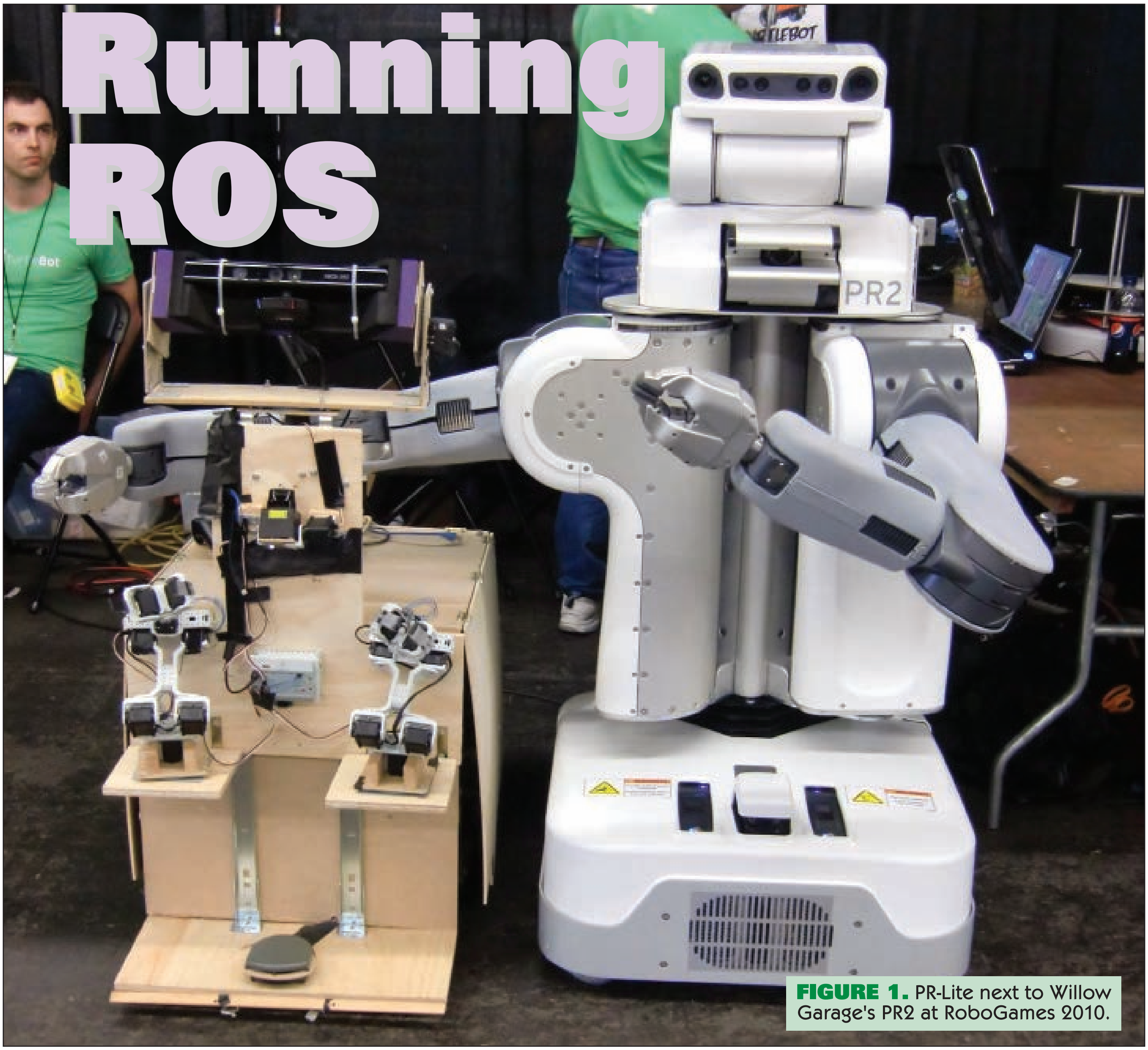


FIGURE 1. PR-Lite next to Willow Garage's PR2 at RoboGames 2010.

by Alan, Andrew, and Matthew Downing, Nathaniel Lewis, and Frank and Robert Ou

How would you like a robot that can play pool, fold towels, and bring you a drink from the fridge? We'd like one too, but don't want to spend $400,000 for a PR2 from Willow Garage. PR2 is built with high-end components so that research on robotics software can progress in anticipation of ultra-low cost components becoming available down the road.

In many ways, that day has already arrived, as embodied by the $150 Kinect and the low cost LIDAR used by the Neato robotic vacuum cleaner. These new sensors and servos with feedback — like the Dynamixel AX-12+ servos — enable a personal robot with PR2-like capabilities to be within reach of a robotics club. Furthermore, the Robot Operating System (ROS) software provides much of the intelligence and glue to make this desire a reality. Such was our goal when we combined the resources and capabilities of four high school roboticists and their mentor fathers to produce PR Lite.

Comparison With PR2

As shown in **Table 1**, the capabilities of PR Lite are analogous to specifications of the PR2, but are lower quality.

	PR Lite	PR2
Head Camera	Kinect (two cameras with LED texture projector for 3D vision)	Three cameras with LED texture projector for 3D vision
Base LIDAR	Neato	Hokuyo UTM-30LX
Tilting LIDAR	Hokuyo URG-04LX-UG01	Hokuyo UTM-30LX
Arm DOF	5 (shoulder pan, tilt, elbow, wrist rotate, left finger, right finger)	7 (includes forearm and upper arm roll)
Arm Payload	2.5 lbs	4 lbs
Number of Arms	Two	Two
Drive System	Forward with differential steering; spin in place; sideways with differential steering	Holonomic
Torso	12" telescopic	12.3" telescopic
Computer	i7 quad-core with hyper-threading, 6 GB memory, 60 GB solid-state drive	Two i7 quad-core, 24 GB memory, two TB disk space (all 2x)
Network	Eight USB ports, I²C	EtherCAT
Batteries	Five 12V lead-acid; 30 min runtime	Lion; Two hour runtime
Software	ROS	ROS
Joystick Control	Yes	Yes
Wireless	Yes	Yes
IMU	SparkFun six DOF*	Microstrain 3DM-GX2
Arm Cameras	Webcam*	Ethernet camera

* Part currently not installed on robot.

TABLE 1.

Open Source Hardware and Software

Our team experience has proven valuable for implementing the initial version of PR Lite. We hope that our efforts will make it easier for others who try to build their own PR Lite. Like with PR2, we embrace the open source philosophy. Our software, schematics, URDF (Universal Robot Description Format) definitions, and CAD drawings are all available online. All other parts are commercial off-the-shelf (COTS) hardware. The other key software is ROS which is open source. PR Lite's PC and AVR code are in online Git version control repositories hosted by Github.

ROS

ROS contains a plethora of functionality that continues to expand due to a large open source community. At its heart, ROS is a message passing system which allows robotics software to be more modular and reusable by defining and allowing the community to define standard interfaces. As a result, many robot components now have ROS interfaces. For example, ROS has Simultaneous Localization And Mapping (SLAM) capability that can be used by both PR2

FIGURE 2. Robert Ou, Andrew Downing, Matthew Downing, and Nathaniel Lewis work on software and CAD drawings for PR Lite.

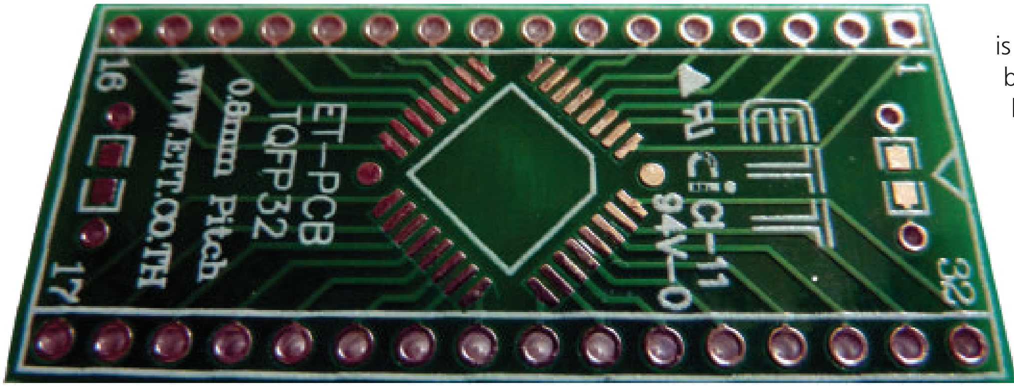


FIGURE 3. Example of breakout board used with the ATmega chips.

and other robots. ROS has drivers for Hokuyo LIDARs, AX-12 servos, the Sparkfun IMU, and many other hardware devices. To help overcome a large learning curve, ROS has many tutorials to explain the functionality available, but these tutorials take time to run, understand, and retain.

After getting over a significant initial learning curve, we were able to easily integrate the Kinect and the Neato LIDAR soon after their release. The Kinect uses the OpenNI library and PrimeSense's NITE library to find a subject's skeleton. This skeleton is transformed into spherical coordinates (with a radius), and inverse kinematics are used to position PR Lite's arms. ROS packages from the University of Arizona for their CrustCrawler arms are customized to control the AX-12+ servos for our very similar arm configuration. Existing drivers for the Hokuyo and Neato LIDARs are used.

PR Lite's ROS navigation stack is customized to handle its unusual base controller to emulate holonomic drive and use the Parallax odometry information. The pr2_teleop_general driver is modified so that PR Lite can be fully controlled by a PS2 joystick using an aftermarket adapter that can easily be found online (rather than the PS3 joystick using Bluetooth).

The ROS simulation software also has a large learning curve. Manipulating XML files to create the robot model is tedious. Our robot is still undergoing significant transformation, so we decided to concentrate more on the design and building of the physical robot. While in theory, the simulation allows software development to continue in parallel with modifying the physical robot, the amount of time to get meaningful results from the simulation is almost as high as getting the physical robot up and working. We eventually put our simulation work on hold so that we could complete our demo for RoboGames.

In retrospect, the amount of software that we wrote to get PR Lite to perform various complex capabilities such as navigation and mimicking human arm movements is surprisingly small. However, the amount of software that we had to understand to get to this point was huge. In the longer term, we'd like to be able to reuse higher level code to perform PR2 demonstrations, but we'll have to modify many of the PR2 services to generalize numerous PR2-specific assumptions.

Hardware Overview

PR Lite is actually a conglomeration of some of our previous robots. Our RoboMagellan robot — which won silver at RoboGames 2010 — was gutted for parts including its two pairs of Parallax wheel kits, HB-25 motor controllers, and its sensors. A Bioloid comprehensive kit

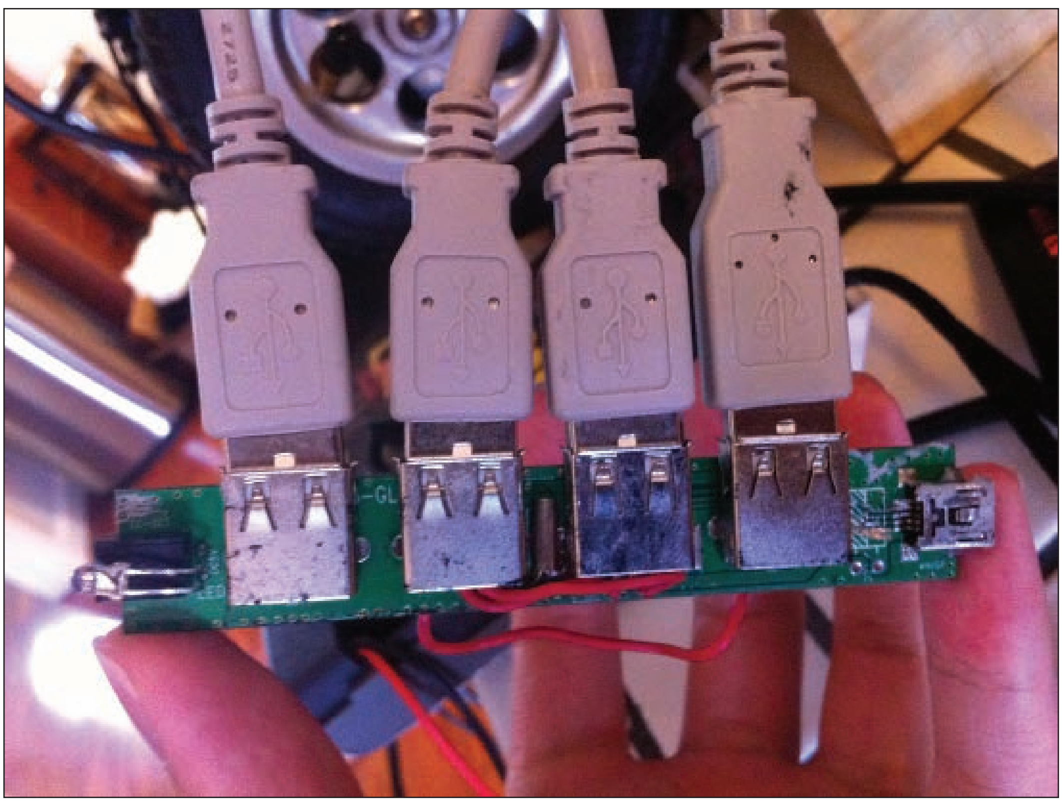
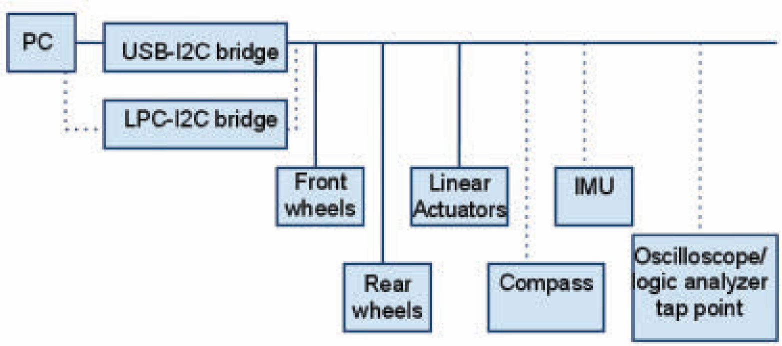


FIGURE 4. USB hub that was modified to connect I²C devices.

FIGURE 5. Devices that are attached to the I²C communications bus (dotted lines represent devices that are not currently installed).



was repurposed from a newer incarnation of Pooh Bear Bot. The Dynamixel AX-12+ servos have position, temperature, voltage control, and feedback, and are used for arms and pan/tilt functionality. A LIDAR was removed from a used Neato that was purchased on eBay. Our RoboMagellan robot was controlled by an OLPC XO-1 which performed poorly when task switching and experienced high latency when accessing USB devices. Learning from our mistakes, PR Lite is equipped with an i7 quad-core processor and other spare computer parts to make our high-end "Frankenstein computer."

PR Lite is designed to offload all timing-critical functions such as PID speed control loops, linear actuator controllers, and the emergency stop capability onto microcontrollers rather than using the Linux kernel real time capabilities like PR2. All our microcontrollers use the ATmega328p because it is sufficiently powerful, used in the Arduino, and has hardware support for I²C. The TQFP surface-mount version is used because we intend to manufacture custom PCBs. Cheap TQFP breakout boards are used for prototyping.

To interconnect the microcontrollers and the PR Lite computer, we chose I²C. However, instead of using I²C in the traditional device and register protocol, it is used to transmit packets of data like UDP. Because the I²C repeated start condition effectively allows a node to hold the bus — forcing other nodes to wait — support for request-response packets is also added.

We utilize the wire-OR nature of I²C to construct a low latency multi-master bus. All the microcontrollers can transmit broadcast or unicast packets onto the bus at any time. The wire-OR allows collision detection, and packets are re-transmitted. The bus runs at 400 kHz.

Each packet has a checksum. If the checksum is wrong, the receiver will discard the packet. The transmitter should then re-transmit the packet, but this has not yet been implemented. The theoretical latency of this bus is in the range of milliseconds which was considered low enough.

I²C operates at the TTL level and is susceptible to noise. We place all the microcontrollers close together with short wires to reduce the noise interference. We use modified USB cables which are shielded for this purpose. We modified a USB hub to be our I²C bus hub by removing the USB hub IC from it and soldering wires to form an I²C bus inside the USB hub chassis. The noise interference is low with this construction but the USB hub IC is surprisingly fragile and unreliable.

To connect the i7 central computer to the microcontrollers, we used a USB-to-serial module to connect to yet another microcontroller — the bridge controller. This bridge controller takes one mega baud serial from the PC USB module and converts the commands to the modified I²C bus for all other microcontrollers. The bridge controller also receives all the status from the microcontrollers and relays them back to the PC.

Drive System

The Parallax wheels are distance controlled. These needed to be modified to be velocity controlled to be compatible with ROS. The Flash in the microcontroller embedded in the Parallax wheel is near full, so we added our own faster microcontroller with larger Flash for velocity and PID control. To utilize the existing Parallax wheel encoders, we did not remove the wheel controller board. Instead, wires are soldered onto the existing optical encoders and connected to our own microcontroller board, bypassing the Parallax controller.

We use two microcontrollers to control four wheels; one for the front wheels and one for the back wheels. Each wheel is individually controlled by the central i7 computer. The long wires from the optical encoders pick up noise from the motors and can cause problems for our microcontrollers. We use optoisolators to isolate the encoder wires from the microcontroller digital inputs.

The two linear actuators are controlled by one ATmega328p microcontroller. We use two relays for controlling each of the two linear actuators. The two relays allow the microcontroller to provide positive 12V, negative 12V, or no power to the linear actuator. The microcontroller can control the linear actuator to move in forward direction, in reverse direction, or to stop.

The linear actuator has a variable resistor as a feedback mechanism. The variable resistor is connected to the input of the microcontroller's built-in analog-to-digital converter. By reading the A/D value, the microcontroller software constantly monitors exactly where the linear actuator is positioned. When the i7 central computer programs or commands the linear actuator microcontroller to move the linear actuator to a desired position, the microcontroller drives the relays to provide proper power to the linear actuator. When the linear actuator reaches the desired position, the microcontroller tells the relays to provide no power and thus stop the linear actuator at the desired

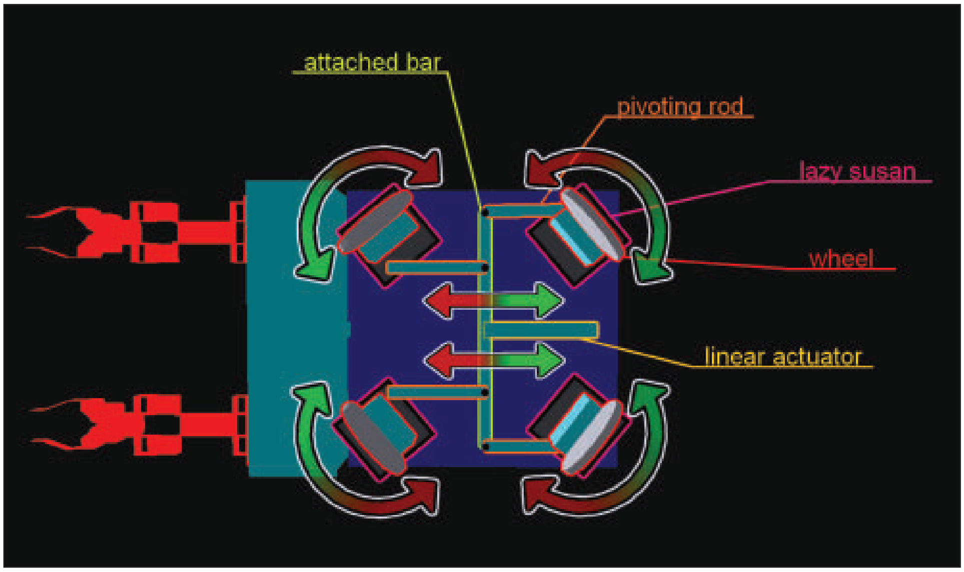


FIGURE 6. Diagram of the mechanism used to rotate the wheels.

position. Unfortunately, the relays and the linear actuators generate a lot of electrical noise. To help reduce this, the control of the relays from the microcontroller is opto-isolated. A separate DC/DC power circuit is used to provide power to the relays.

Many different power adapters are needed to power different components. Originally, we tried to build DC/DC power sources for each voltage type. Later, we decided to just use AC power for many of the components via a 200W DC-to-AC inverter. We can connect the battery to this inverter and allow many components to run with their original AC power adapters. The i7 computer runs on its own 24V DC power supply, which is powered by two 12V lead acid batteries.

The Base and Torso

We used wood for the base of our initial PR Lite prototype. Wood is cheap and easy to cut and drill with hand tools, allowing us to tweak the design continuously. The base contains a bottom shelf with a tongue that extends beyond the main cabinet and holds the Neato LIDAR. A telescoping torso is attached to the front side of the cabinet.

Inside the main cabinet are three shelves. The bottom shelf contains the rotating wheels, a linear actuator, and batteries. The top and middle shelfs hold the computer, all the microcontrollers, the DC-DC and DC-AC converter, and the Wi-Fi router. The Parallax wheels are encased in acrylic structures modeled with CAD software which are each attached to a 4" lazy susan. The four wheels are all simultaneously pivoted by a single bar attached to a linear actuator. The linear actuator has a 4" throw with feedback and is controlled by a microcontroller. An H-bridge is formed via relays and transistors. As illustrated in **Figure 6**, the wheels are pivoted in place to one of three positions for the wheels to move forward, pivot in place, or move sideways. When moving forward or sideways, differential drive is used for further turning control. The wheel controller will not simultaneously move the wheels while pivoting them via the linear actuator.

The torso is attached to the base via two 12" drawer slides and can be moved up and down using a linear actuator. Attached to the top of the torso is a Kinect that can pan and tilt through a pair of servos and three lazy susans. The arms are made from a Bioloid comprehensive kit and are constructed to be similar to two Crustcrawler arms. Each Bioloid arm is attached to a shelf on the torso via a lazy susan which is rotated by a servo. A 4" lazy susan provides a stable base that can handle a heavy load so that the servo only has to deal with the rotational force. The servo hub is centered within the hole of the lazy susan and attached to a shelf on the telescoping torso.

Because the AX-12+ wires are daisychained, one wire with poor connectivity can make the whole downstream chain perform poorly or not function at all. The wires in our Bioloid kit are modified to be customized lengths. Many needed replacement until we produced reliable arm performance.

Conclusion

Our completed PR Lite prototype was demoed at RoboGames 2011 and featured PR Lite's arms mimicking a person's arms. Many enhancements are planned to make PR Lite even more like PR2 in both dimension and capabilities. The main addition will be to add upper arms, each with pan capabilities (using two lazy susans with dynamo servos) and powerful lift capabilities (using a linear actuator). Using our many lessons learned, our next iteration will be fully sketched out using CAD software and made from laser-cut ABS at the TechShop. We also want to harden the prototype against loose wires and bolts by fabricating some boards, and replacing many of the Bioloid construction kit parts with simple bent aluminum bars.

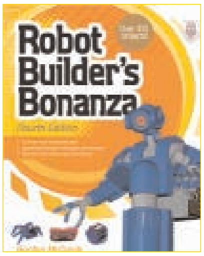
Much of the remaining work is software. The robot torso will be adjusted so it's similar in shape and dimensions to PR2, allowing us to more easily reuse code intended for PR2. We will try to hook PR Lite arm planning directly into the PR2 arm planning. We'll revisit the simulation for the next iteration of PR Lite (PR2 Lite). We can envision PR Lite eventually being able to perform many PR2 capabilities. We look forward to doing our own hackathons! **SV**

The *SERVO* Webstore

Attention Subscribers ask about your discount on prices marked with an *

ROBOTICS BOOKS UNDER $35 — *GREAT GIFTS FOR DIYers!*

Robot Builder's Bonanza, Fourth Edition
by Gordon McComb



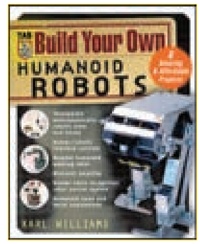
Robot Builder's Bonanza, Fourth Edition includes step-by-step plans for a number of motorized platforms. The book is described as a compendium of robotics topics, containing more than 100 projects, including 10 robot designs new to the fourth edition. These modular robots are low cost, and are made to be reproduced by readers with no training in mechanical construction.
$29.95*

Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists
by Dustyn Roberts

In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects — from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.
$29.95*

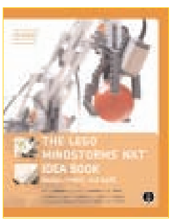
Build Your Own Humanoid Robots
by Karl Williams

GREAT 'DROIDS, INDEED!



This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot. **$24.95***

LEGO MINDSTORMS NXT Idea Book
by the Contributors to The NXT Step Blog



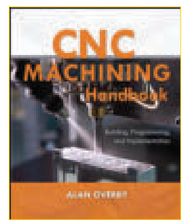
If you're serious about having fun with LEGO® robotics, you've come to the right place. The team behind The NXT STEP blog — the authoritative online source for MINDSTORMS® NXT information and advice — has packaged its considerable skills and experience in this book. Inside, you'll find some of the team's best ideas for creating cool and sophisticated models, including instructions for eight robots you can build yourself.

Reg $29.95 **Sale Price $24.95**

RobotBASIC Projects For Beginners
by John Blankenship, Samuel Mishal

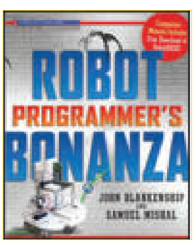
If you want to learn how to program, this is the book for you. Most texts on programming offer dry, boring examples that are difficult to follow. In this book, a wide variety of interesting and relevant subjects are explored using a problem-solving methodology that develops logical thinking skills while making learning fun. RobotBASIC is an easy-to-use computer language available for any Windows-based PC and is used throughout the text.
Reg. Price $14.95 **Sale Price $9.95**

CNC Machining Handbook: Building, Programming, and Implementation
by Alan Overby



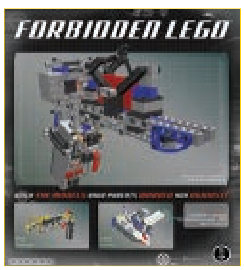
The *CNC Machining Handbook* describes the steps involved in building a CNC machine and successfully implementing it in a real world application. Helpful photos and illustrations are featured throughout. Whether you're a student, hobbyist, or business owner looking to move from a manual manufacturing process to the accuracy and repeatability of what CNC has to offer, you'll benefit from the in-depth information in this comprehensive resource. **$34.95**

Robot Programmer's Bonanza
by John Blankenship, Samuel Mishal



The first hands-on programming guide for today's robot hobbyist!
Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks. **$29.95**

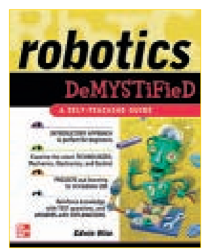
Forbidden LEGO
by Ulrik Pilegaard / Mike Dooley



Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse. **Reg $24.95** **Sale Price $19.95**

Robotics Demystified
by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!



Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **$19.95**

For complete product details, visit our webstore online!!

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

The NXT Big Thing #15

Discuss this article in the *SERVO Magazine* forums at http://forum.servomagazine.com

By Greg Intermaggio

Konichiwa, arigato! Wait. That's not right. Hello, thank you? Hmmm ... Anyway, welcome back to The NXT Big Thing! Since our last edition, I've been to and fro from the land of upside down sheep ... New Zealand!

That said, this month we'll be building a fantastic, custom robot chassis called Isotope. Isotope is similar to other robots in that it has most of the same components. What makes it Isotope are the two linear actuators from Firgelli which are powered by the NXTMMX from MindSensors, that allow the robot to extend and retract its front wheels. This ability is both awesome (it looks a lot like cars with hydralics) and functional (it allows Isotope to maneuver difficult terrain).

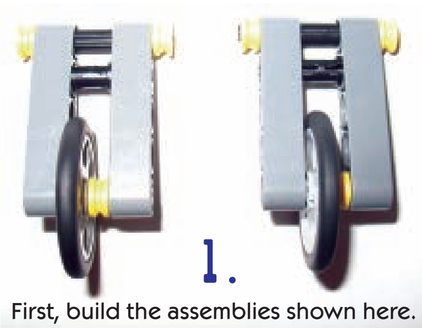
This month, we'll be putting together Isotope and writing a very basic program to test it. Next month, we'll use a second NXT to create a Bluetooth controller for Isotope which will make it easy to show off his awesome functionality in any situation.

To purchase Firgelli linear actuators, visit **Firgelli.com**.

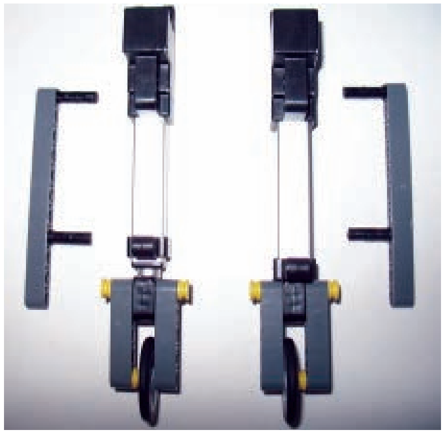
To purchase the NXTMMX and its battery pack, visit **MindSensors.com**.

Now, let's get started!

Building Instructions: Isotope



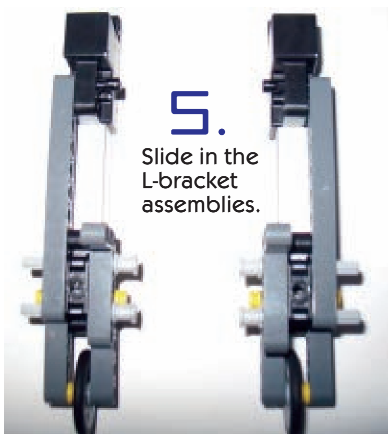
1. First, build the assemblies shown here.



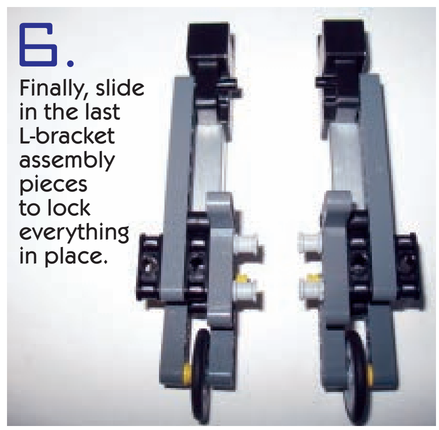
2. Attach them to your Firgelli linear actuators as shown, then grab two 11-hole studless beams and four double friction pins, and prepare them to attach.

3. Attach your pins and studless beams to the linear actuators.

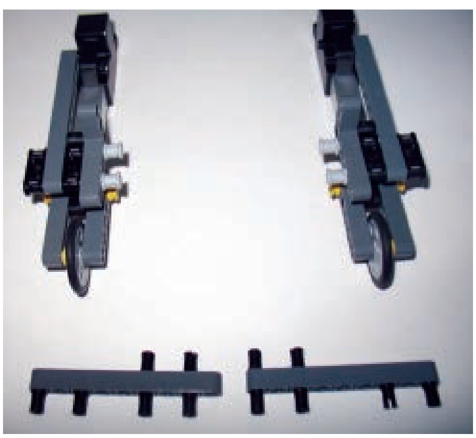
4. Prepare the L-bracket assemblies shown which will help provide support to Isotope.



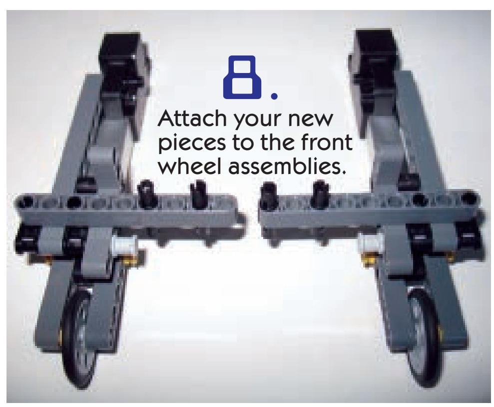
5. Slide in the L-bracket assemblies.



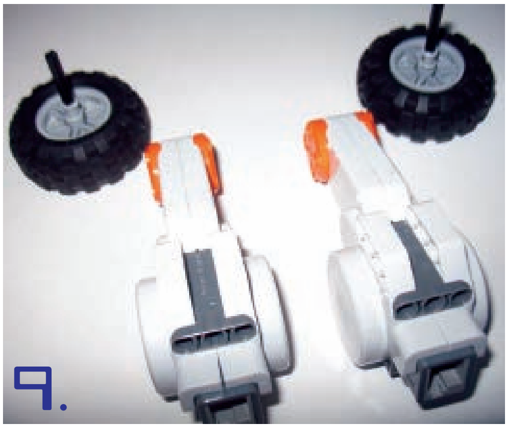
6. Finally, slide in the last L-bracket assembly pieces to lock everything in place.



7. Collect two nine-hole studless beams, four double friction pins, and four standard friction pins. Attach them as indicated. You'll notice that one side is off center. This is intentional, and you'll find out why shortly!

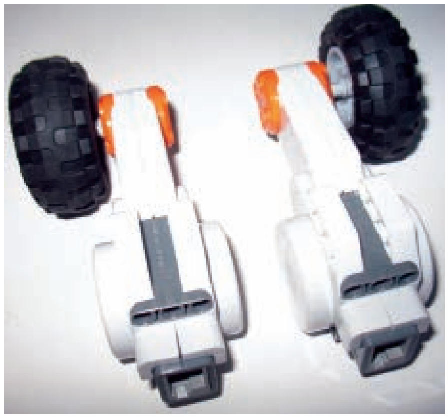


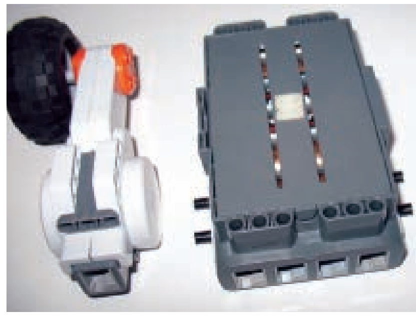
8. Attach your new pieces to the front wheel assemblies.



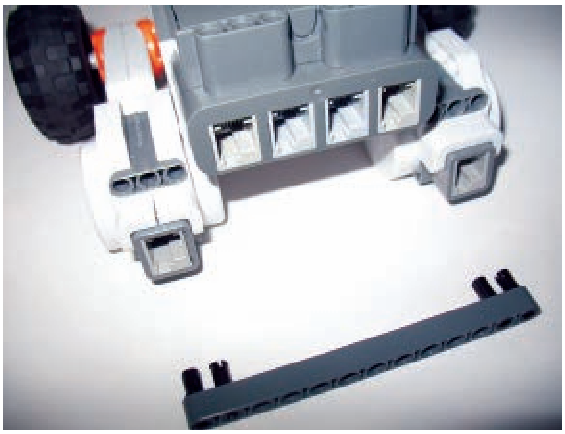
9. Now, grab two motors, two wheels, two bushings, and two six-stud axles. Slide the axles through the wheels and put the bushings on.

10. Pop the wheels onto the motors.

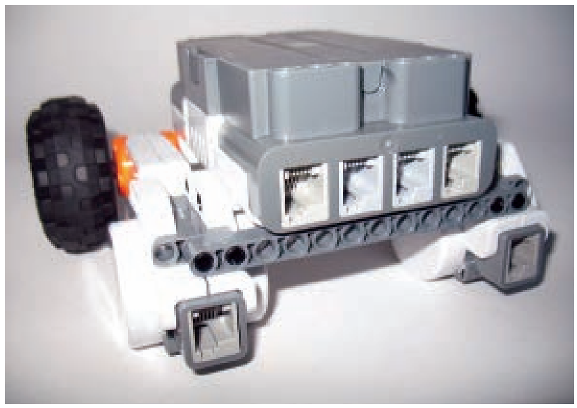




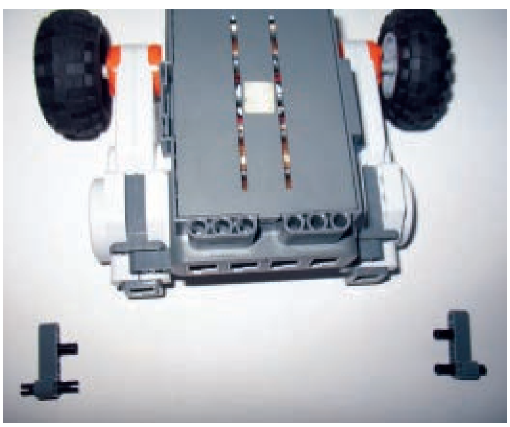
11. Now, grab an NXT and flip it upside down (this model is compatible with both AA batteries and rechargeable batteries powering the NXT). Snap two friction pegs on each side of the NXT as shown, and prepare to attach the motors to them.



12. Snap in the motors, then grab a 13-hole studless beam and four friction pins; prepare them as shown.

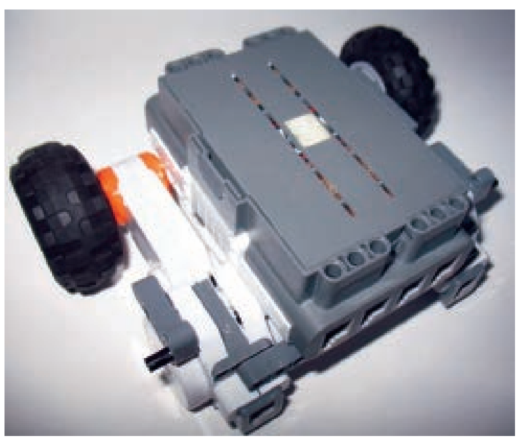


13. Now, pop the beam and pegs in place.

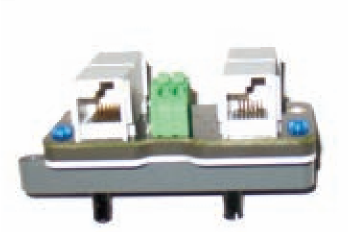


14. Grab two small L-brackets, two friction pins, and two double friction pins, and snap 'em together as indicated.

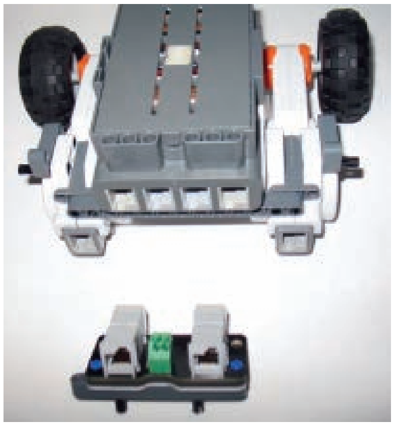
15. Snap one on either side of the NXT as shown.



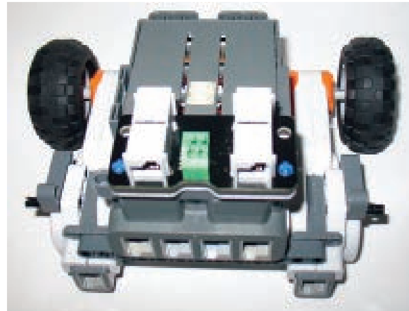
16. Grab your NXTMMX, a nine-hole studless beam, two friction pins, and two axle-pins. Snap the pins into the beam as shown.



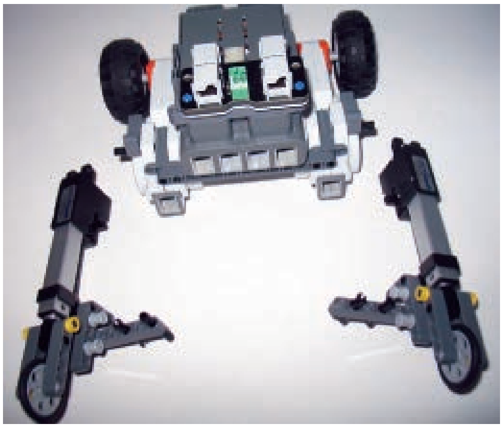
17. Push the axle pins into the NXTMMX. This might take some force, so be careful!



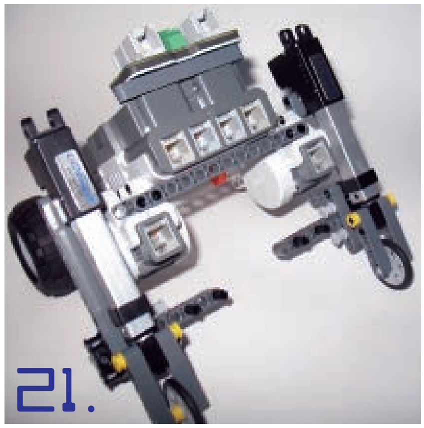
18. Grab your NXTMMX assembly and your main chassis.



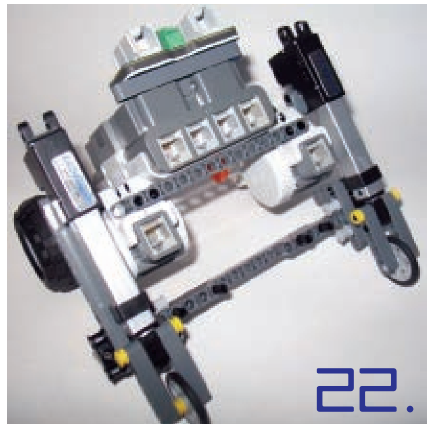
19. Snap the NXTMMX to the main chassis as shown.



20. Grab your two front wheel assemblies. Note their orientation in this picture (L-brackets pointing down).

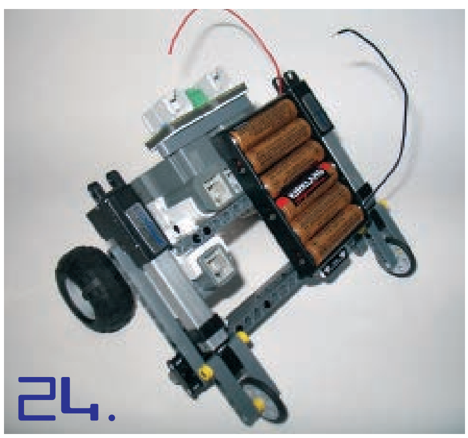


21. Snap your Firgelli actuators to the small L-brackets on the main chassis. The two holes on each actuator should line up with the two holes on each small L-bracket.



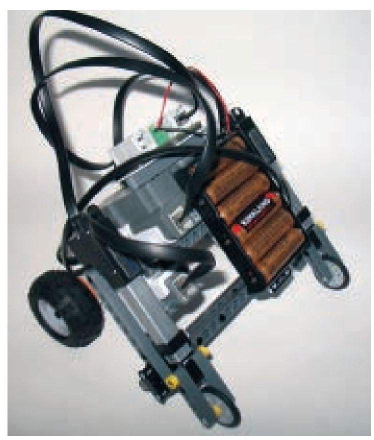
22. Run a 13-hole studless beam across the two front wheel assemblies.

23. Snap another 13-hole studless beam on top, with two axle extender pins on one side as shown. The studless beams will reinforce Isotope's wheels so they don't wander.



24. Attach the NXTMMX battery pack using two axles to reach the axle extender pins, and two axle pins (obscured in this image by the battery pack) to reach the other support beam to create four total points of attachment.

25. Plug the battery pack wires into the NXTMMX. Plug the drive motors into ports B and C; the Firgelli linear actuators into the NXTMMX motor ports; and plug the NXTMMX into the NXT in sensor port 4.



Now that we've got Isotope built and ready to go, let's get him programmed! Before you program Isotope, you'll need to download and import the NXTMMX block from **MindSensors.com**.

Give your program a test. If all goes well, Isotope should extend his front wheels fully, then drive forward, then retract them. Awesome!

That wraps up this month's edition of The NXT Big Thing. Stay tuned for the next one! **SV**

Isotope Program Instructions

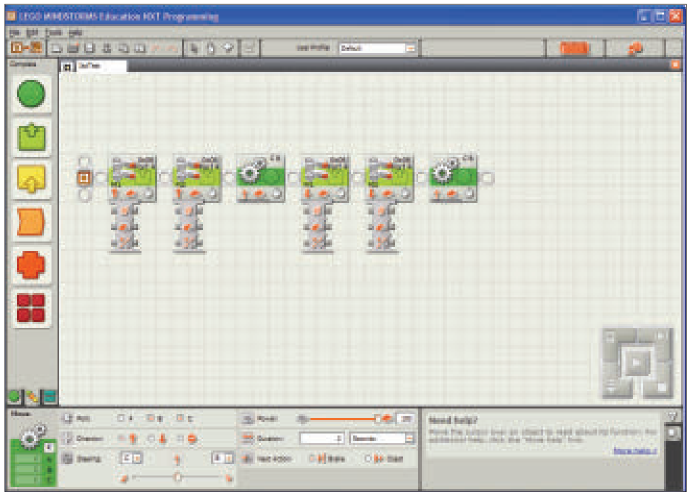


Figure 1. After downloading and importing the NXTMMX block from MindSensors, simply drag in two NXTMMX blocks, a Move block, another two NXTMMX blocks, and another Move block.

- Set all the NXTMMX blocks to port 4, and all of the motor blocks to ports B and C.
- Set the first NXTMMX block to Motor 1 forward for four seconds, and the second to Motor 2 forward for four seconds.
- Set the first Move block to B and C forward for two seconds.
- For the third and fourth NXTMMX blocks, set them to move Motors 1 and 2 backwards for four seconds, respectively.
- For the final Move block, set the direction to forward and the duration to five seconds!

THE OWNERSHIP, MANAGEMENT, AND CIRCULATION STATEMENT OF SERVO MAGAZINE, Publication Number: 1546-0592 is published monthly. Subscription price is $24.95. 7. The complete mailing address of known office of Publication is T&L Publications, Inc., 430 Princeland Ct., Corona,Riverside County, CA 92879-1300. Contact Person: Tracy Kerley. Telephone: (951) 371-8497. 8. Complete Mailing address of Headquarters or General Business Office of Publisher is T&L Publications, Inc., 430 Princeland Ct, Corona, CA 92879. 9. The names and addresses of the Publisher, and Associate Publisher are: Publisher, Larry Lemieux, 430 Princeland Ct., Corona, CA. 92879; Associate Publisher, Robin Lemieux, 430 Princeland Ct., Corona, CA 92879. 10. The names and addresses of stockholders holding one percent or more of the total amount of stock are: Jack Lemieux, 430 Princeland Ct., Corona, CA 92879; Larry Lemieux, 430 Princeland Ct., Corona, CA 92879; Audrey Lemieux, 430 Princeland Ct., Corona, CA 92879; 11. Known Bondholders, Morgagees, and other security holders: None. 12. Tax Status: Has not changed during preceding 12 months. 13. Publication Title: SERVO Magazine 14. Issue Date for Circulation Data: October 2010-September 2011. 15. The average number of copies of each issue during the proceeding twelve months is: A) Total number of copies printed (net press run); 11,668 B) Paid/Requested Circulation (1) Mailed Outside County subscriptions: 4,499 (2) Mailed In-County subscriptions: 0 (3) Paid Distribution Outside the Mail including Sales through dealers and carriers, street vendor, and counter sales and other paid distribution outside USPS: 2,057 (4) Paid Distribution by other classes of mail through the USPS: 0; C) Total Paid Distribution: 6,556; D) Free or Nominal Rate Distribution by mail and outside the mail (1) Free or Nominal Rate Outside-County Copies: 0 (2) Free or Nominal Rate In-County Copies: 0 (3) Free or Nominal Rate Copies Mailed at other classes through the USPS: 0 (4) Free or Nominal Rate Distribution Outside the mail: 667; E) Total Free or Nominal Rate Distribution: 667; F) Total Distribution: 7,223; G) Copies not distributed: 4,445; H) Total: 11,668; Percent paid circulation: 90.76%. Actual number of copies of the single issue published nearest the filing date is September 2011; A) Total number of copies printed (net press run) 10,440; B) Paid/Requested Circulation (1) Mailed Outside County subscriptions: 4,332 (2) Mailed In-County subscriptions: 0 (3) Paid Distribution Outside the Mail including Sales through dealers and carriers, street vendor, and counter sales and other paid distribution outside USPS: 1,959 (4) Paid Distribution by other classes of mail through the USPS: 0; C) Total Paid Distribution: 6,291; D) Free or Nominal Rate Distribution by mail and outside the mail (1) Free or Nominal Rate Outside-County Copies: 0 (2) Free or Nominal Rate In-County Copies: 0 (3) Free or Nominal Rate Copies Mailed at other classes through the USPS: 0 (4) Free or Nominal Rate Distribution Outside the mail: 500; E) Total Free or Nominal Rate Distribution: 500; F) Total Distribution: 6,791; G) Copies not distributed: 3,649; H) Total: 10,440; Percent paid circulation: 92.64%. I certify that these statements are correct and complete. Larry Lemieux, Publisher.

Then and NOW

A ROBOT IN EVERY HOME?

b y T o m C a r r o l l

I have a floor to ceiling bookcase in my office that is filled with books on robotics. Many of the titles are from the '70s and '80s. One in particular caught my eye recently — "A Robot in Every Home" by Mike Higgins, published in 1985 by Kensington Publishing Company. When I bought this book back in the mid '80s, I was thinking that this possibility is just way too cool.

The cover illustration shown in **Figure 1** depicts the 'man of the house' seated in a chair, reading a book with his wife dutifully standing behind him. Beside him you can see the wheeled humanoid robot holding the couple's smiling baby with one hand while holding a baby bottle in the other. Remember, this is 1985 when a 'home' robot was probably a Topo, Hero 1, Hubot, RB5X, or maybe an Omnibot — none of which any sensible and caring parent would trust handling their infant child. So, where are these home robots today?

A Personal Assistant Robot

The idea of an all-purpose home robot in my home is a thought that I have considered quite carefully these past few months. In the past, a home robot would have been used as a research platform for any type of experiment I chose to perform. However, I recently had a prostatectomy; the daVinci robot-assisted operation was quite successful. I will cover the whole experience in a future article, but the vision of a truly useful personal assistant robot has become a more vivid need to me, as well as for the millions of others who might need a bit of assistance in daily living.

A recent article by Krystal D'Costa from the *Scientific American* blog site touted: "A Robot in Every Home? We're Getting Close — Will we recognize our robot overlords when we meet them?" Now, that's something to think about. What does the statement: 'A robot in every home' actually mean? Will people perceive robots as some sort of 'overlord?' Let's face it, there will never be a robot in every home — not here in the US or even in Korea where that was the unofficial mandate a few years back. There is not even basic plumbing or electricity in every home. That statement really meant that robots would be available for the vast majority of people, not our entire populace.

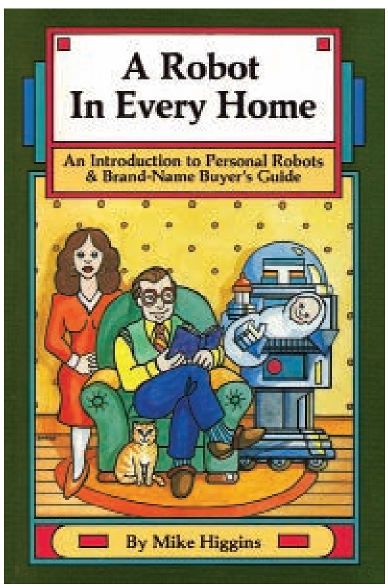


FIGURE 1. Mike Higgins' *A Robot In Every Home.*

People shunned microwave ovens, televisions, cars, telephones, and many other 'technological wonders' in the beginning of their existences but soon learned to welcome them. The music industry laughed at MP3 players as a silly fad until the sales of CDs took a dramatic slump. If a certain technology will make parts of life easier for us humans, it will be accepted.

The Reality of a Home Robot

Is the thought of a robot in every home just a pipe dream by a few individuals who do not have a clue of what technology is involved to produce a viable home robot? I am quite sure that there are some clueless writers who don't have any understanding of the complexity of robotics. However, there are some people that speak or write about robotics in the home who have done their research, and

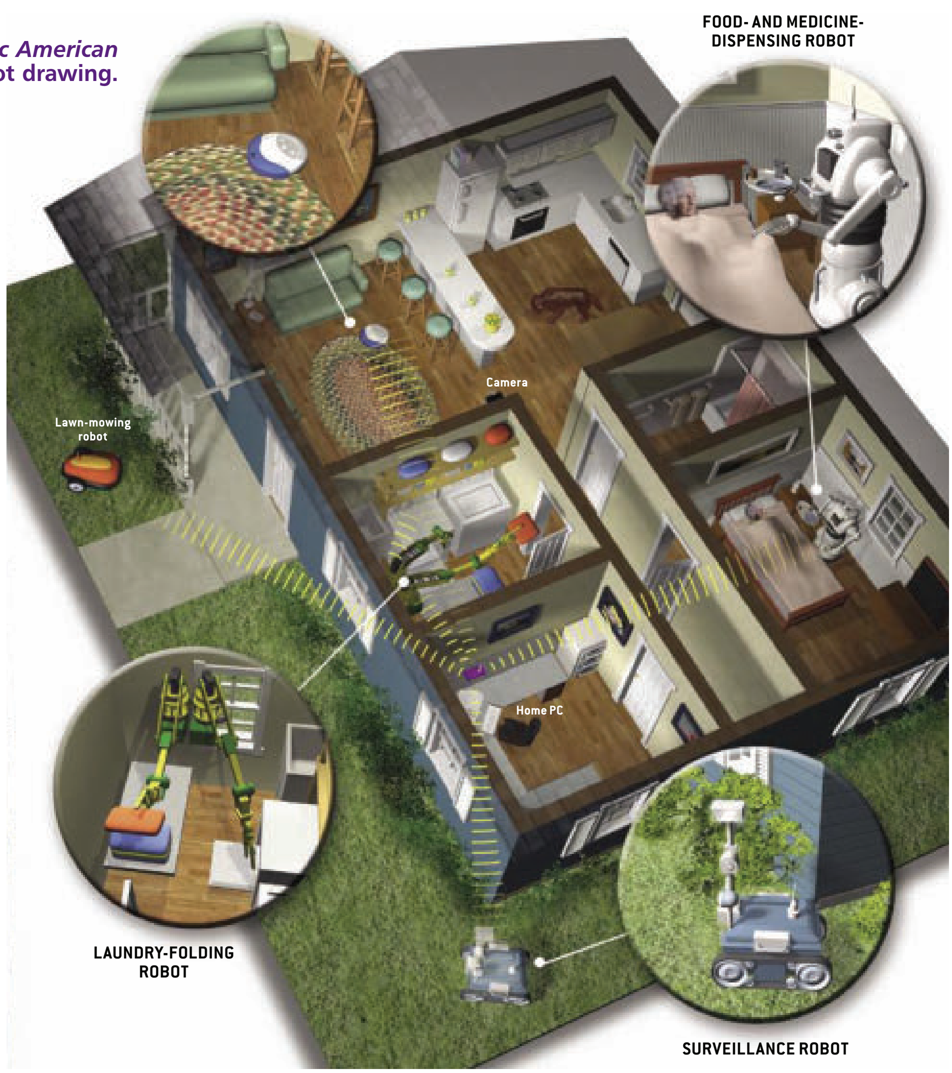


FIGURE 2. Bill Gates' '06 *Scientific American* home robot drawing.

they have some fairly sound futuristic predictions. Five years ago, Bill Gates wrote a very introspective article about a robot in every home in the December '06 *Scientific American* and was attacked by many people for months afterwards about his comments. It wasn't because Gates was incorrect; he was pretty much right on the money with his futuristic opinions, though some of the robots shown in **Figure 2** were a bit too specialized. He was attacked because he was the software baron of the world and was therefore ripe for any disparaging comments. His article served as a stepping stone for many articles and conversations regarding whether we could have, not necessarily will have, or should have a robot in every home.

What is the Best Type of Home Robot?

I've probably opened a flood gate of opinions with that question. Bill Gates' drawing in **Figure 2** was not depicted as a typical ideal home layout but just a representative house with at least one of each of the home features. From the older lady shown in the bedroom, this home appears to be occupied by a single senior.

Let's start with the floor cleaning robot pictured at the top left. It appears to be an earlier version of iRobot's Roomba. Here at the end of 2011, over six million Roombas have been sold world-wide — by far the most popular of all floor-cleaning robots. I've had two different Roombas over the past half dozen years, and agree they have their place in today's homes. They do require more hands-on upkeep than a typical vacuum cleaner, so this could be something a senior may not want to do.

A Personal Assistant Robot in the Home

This type of robot design has been near and dear to me for 20 years, but I have extreme reservations about having a large humanoid robot with two articulated arms that's designed just to dispense food and medicines. So many groups in this country and elsewhere have spent millions of dollars trying to develop a sophisticated home robot, only to end up with a roving tele-operated machine that can navigate fairly successfully, but do little else other than visual monitoring and verbal communications, such as: "Grandma, it's time to take your medicine. Please remove the yellow pill from my box and take it with a glass of water." Do you really need a mobile robot to accomplish this simple task? **Figure 3** is an early depiction of Joe Engelberger's Isaac home robot, developed by his Helpmate Robotics company. Note the similarity of the base to his earlier hospital robot.

I've always felt that the best personal assistant robot would be that — an assistant to a person who needs assistance. Not a simple errand boy that just retrieves needed food, medicines, and objects when requested, but a physically capable robot that can assist a person desiring an independent living capability. A robot that can gently and safely assist a person out of and back into bed, on and off the toilet, and assist them from the floor after a fall is what so many seniors desire.

"Pride in independent living without the fuss and bother of a person hanging over me all day is what I want." That's a comment I've heard hundreds of times when talking with senior groups about some of my robot designs.

The Surveillance/ Security Robot

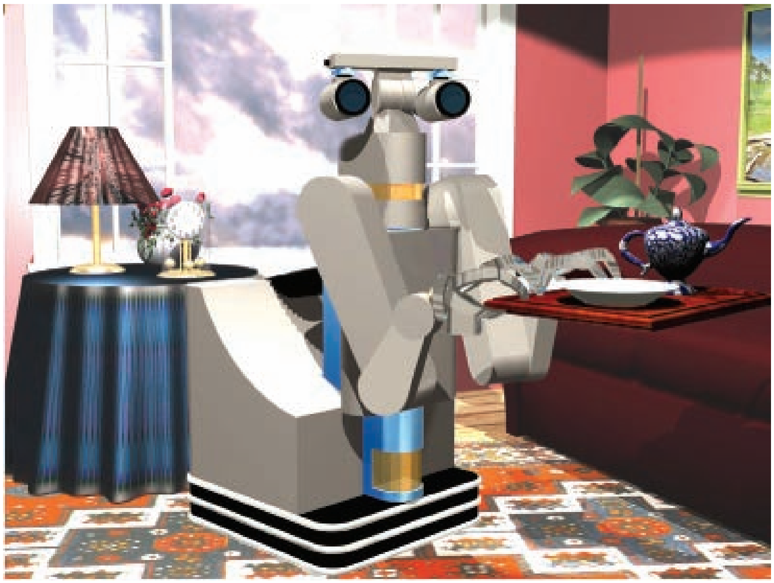


FIGURE 3. Joe Engelberger's home robot.

At the bottom of **Figure 2** a surveillance robot is depicted. My first impression of such a robot roaming about the outside of someone's home is that somebody is going to steal that thing! Some security! Why have a mobile security robot? Why not just augment a nice monitored home security system with extra TV cameras? Leave the high-tech mobile robots to warehouse and military weapons surveillance.

The "Simple" Task of Folding a Towel

One type of robot that jumped out at me from **Figure 2** was the laundry folding robot. This task is not as loathed as toilet cleaning, but a few people enjoy the task of folding laundry. Last year, as Willow Garage's PR-2 made the circuit of YouTube videos, one of the most interesting was the $400K robot folding laundry. What many people do not realize is that it took a lot of programming, plus the actual folding operation was a lot longer than shown in the videos. Willow Garage was in no way trying to sell this amazing robot as a laundry room attendant. They were simply illustrating how the robot's vision system can work in conjunction with the two articulated arms to perform a complex task.

Researchers at UC Berkeley developed a cloth-grasping algorithm to be used by their PR-2 to allow it to grasp, pick up, and fold towels it hadn't previously analyzed. **Figure 4** is a photo of the experiment that was written up in a 2010 ICRA paper and included the accompanying video I mentioned earlier. The test involved 50 separate towels and a pile of five. The robot had an 81% success rate in grasping the towels, and took an average of almost 25 minutes to fold each one. The greatest amount of time was required for the PR-2 to just determine where to grasp each towel. (Can you imagine how difficult it would be for this very sophisticated robot to grasp inside-out socks, various sizes and types of underwear, wrinkled shirts, or lingerie and correctly fold them?)

FIGURE 4. Willow Garage's PR-2 towel-folding demo at UC Berkeley.

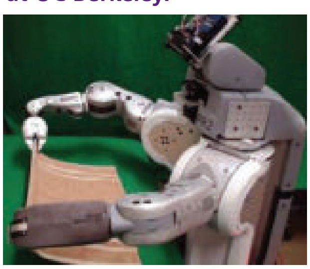


FIGURE 5. The Qbo robot from The Corpora.

Specialty Tasks Within the Home

I started out this article by commenting how on-the-money Gates was in his article and then managed to find problems with each of his examples. All of the types of robots that he mentioned are viable concepts and many are high on the list of people's needs for a robot in the home. It seems that simple tasks for a human are complex tasks for a robot.

If one adds up the unpleasant chores of toilet cleaning (20 minutes a week) and laundry operation for a single person (10 minutes loading and starting the washing machine; five minutes transferring loads from the washer to dryer, and 25 minutes removing, folding, and stowing), you have an hour of menial human labor costs. Add to that the time to do light housecleaning (no Roombas here) and some kitchen prep work, and you might have two to three hours of a human's time. Once a month or so, oven and fridge cleaning can be included, as well as heavier spring cleaning tasks. A lawn service is certainly preferable to a $1,000 lawn robot that is used an hour a week, as is a basic home security system over a mobile robot.

Food and medicine dispensing robots get even more complicated. If the only resident of the home is totally bedridden, then any foreseeable home robot design will have a tough time being accepted by the FDA. The FDA Code of Federal Regulations, part 890.5050, describes a "Daily Activity Assist Device" that might be applicable to most personal assistant robot designs, but regulations are very strict for any 'medical' device that comes in contact with a human. So, how exactly can we make

the best use of robots for the home?

The Ideal Home Robot

Two groups seem to be leading the research on viable home robots. Willow Garage's Scott Hassan (who tapped Steve Cousins as CEO) has made a major splash with the PR-2 — the polished upgrade from the PR-1 developed at nearby Stanford. With this amazing $400K robot at the top end and their $1,400 TurtleBot at the lower end, WG has been the force behind the open source Robot Operating System (ROS). Their applications for a marketable autonomous home robot are being developed with the open source ROS and the Microsoft Kinect sensor system. The much-talked-about open source Qbo 'home service' robot shown in **Figure 5** from the French/Spanish group, The Corpora, has incorporated ROS within its Linux framework.

The other group is the Microsoft Robotics Group, started by early Microsoft employee, Tandy Trower, and a small group of software engineers and robot experimenters. The group's first offering was the Robotics Developer Studio (RDS) in 2006. There were 40K or so users of RDS, but interest waned and Tandy resigned in November '09 to form his own company — Hoaloha Robotics — which launched in 2010 to develop home assistant robots.

Last year, Dr. Stathis Papaefstathiou took over as General Manager of the MS Robotics Group, and his group rapidly developed a new enthusiasm for robotic technology. The RDS4 Beta has recently begun shipping (it's free) and has proven to be a hit — especially with the Kinect for Windows SDK Beta (Kinect SDK) that just released. Not to rest on their laurels, the MS Robotics Group released the free download of Kinect Services for RDS that is built on top of the Kinect SDK. Dr. Papaefstathiou's enthusiasm makes it very clear that Microsoft is in robotics for the long haul.

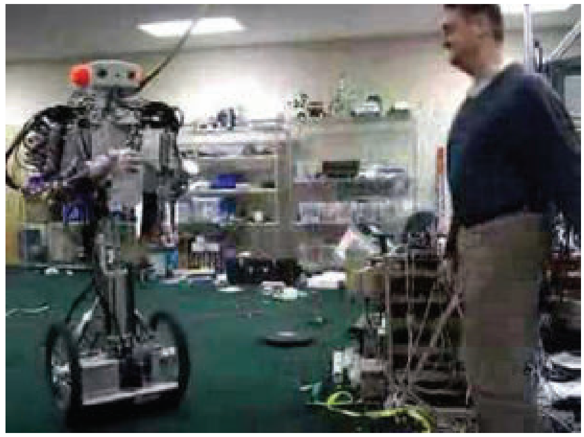


FIGURE 6. Dexter at AnyBots lab.

FIGURE 7. PR-2 concept from April '09.

FIGURE 8. Care-O-Bot assisting a senior.

As he stated in September, "A few years ago, we recognized that the dawn of robotics in the consumer market was coming, and asked ourselves how we could drive the technologies that will make robotics relevant to the consumer market. There is very interesting and exciting work that is taking place in academia, research, community, and among many start-ups. We didn't want to replicate these efforts, so we decided to focus on leveraging Microsoft's vast portfolio of technologies and the brainpower of Microsoft Research to develop capabilities, scenarios, and experiences that are necessary to push robotics over the last "hump" to become part of our everyday lives."

Resources are Available for a New Breed of Home Robot

Home robot designers now have a choice of two mainstream open source software platforms and a very affordable intelligent vision system — Kinect. ROS has been well accepted within the academic community and RDS is favored within many companies and experimenters. These two companies will be driving development of viable home

FIGURE 10. Anybots' Dexter empties a dishwasher.

FIGURE 9. Vecna Technologies' Bear.

robots. Self-balancing, two-wheeled robots such as the Dexter prototype from AnyBots shown in **Figure 6** work well in office and industrial environments but are not really safe enough yet as a full-time home robot. Dual-arm, wheeled humanoid robots such as the WG PR-2 prototype shown in **Figure 7** may be ideal home robot designs but — as proven in sales of the final product — $400K is a bit too much.

A personal robot for the home must be safe around humans, have significant battery life, be able to traverse various home surfaces such as carpeting, doorjambs, objects on the floor, possible slight inclines, narrow doors, and halls, and do so safely, quickly, and quietly. The hands and arms should have sufficient dexterity to handle food items, kitchen utensils, clothing, and — most important — gently but firmly help a human when physical assistance is required.

Figure 8 shows a German concept robot — the Care-O-Bot — assisting a senior. **Figure 9** shows another concept by Vecna Technologies called the 'Bear' — a search and rescue robot designed for the military. While not immediately applicable to the home environment, the paddle hands, base system, and overall design certainly deserve some attention from home robot designers.

Final Thoughts

Most robot experimenters — myself included — would like nothing better than to build the ultimate home robot. That robot may be a simple "Bring me a Pepsi and some chips" type of robot, or maybe a bit more complex type such as "I'd like a two-egg cheese omelet, four strips of bacon, two slices of buttered toast, and a glass of orange juice, and have it at my bedside when I get out of the shower."

Another potential user may want a "Robbie, could you help me out of bed? Call Doctor Smith for me and put him on your speaker phone when he answers."

We will all have our own specific needs in a potential home robot design. Will robots such as these ultimately become companions? You can bet they will. The AnyBots Dexter shown in **Figure 10** may be just a staged photo-op of a robot unloading a dishwasher, but this and other robots have this capability right now. Sophisticated software and sensor packages are here now and are in place on many new robots, and companies like Microsoft and Willow Garage will be developing robots to someday be able to take care of us. The final products should be nothing short of amazing. **SV**

Tom Carroll can be reached at TWCarroll@aol.com.

ROBO-LINKS

Advertiser Index


SERVOCITY
A DIVISION OF ROBOTZONE, LLC.
WE HAVE THE PARTS FOR YOUR IDEAS!
NEW
We are proud to introduce our
BRAND NEW ROBOT KIT!
the ROBOTZONE STACKER
- machined from extremely durable 1/4" ABS plastic
- customizable framework to accomodate various attachments
- R/C or autonomus control*
- large electronics compartment
- omni-directional movement
- additional front mounting plate available
- only weighs 3 lbs
- lifts objects 11.5" when fully extended
- stainless steel caster
- 9.75" H x 7.75" W x 10.25" L
- 3" drive wheels
perfect beginner bot
4.25" gripper spread
Get your Stacker Robot today...kit starting at only
$139.99
(no servos or radio system)
www.youtube.com/user/ServoCity
see videos of the Stacker Robot on ServoCity's YouTube Channel
You Tube
visit servocity.com to learn more about the Stacker Robot Kit
have a smartphone?
scan this code using your smartphone with a bar reader app to watch instant videos of the Stacker Robot in action
To view our entire selection of products visit us online at
WWW.SERVOCITY.COM
*add your own sensors for autonomus control
Like us on Facebook
Check out other product videos
Gears & Sprockets
Shafting & Tubing
Full Line of Hitec & Futaba Servos
Servo Power Gearboxes
Hubs & Adaptors
Copyright 1999-2011 Robotzone, LLC. - All Rights Reserved
ServoCity is a registered trademark of Robotzone, LLC.
Phone: (620) 221-0123
E-mail: Sales@ServoCity.com
www.youtube.com/user/ServoCity
Prices and availability subject to change without notice.